Un modèle multidimensionnel pour un processus d'analyse en ligne de résumés flous

Lamiaa Naoum

UNIVERSITÉ DE NANTES
ECOLE POLYTECHNIQUE

ÉCOLE DOCTORALE STIM

SCIENCES ET TECHNOLOGIES
DE L'INFORMATION ET DES MATÉRIAUX

2006

Thèse de DOCTORAT

Spécialité : INFORMATIQUE

Présentée et soutenue publiquement par

Lamiaa NAOUM

le 22 novembre 2006

au Laboratoire d'Informatique de Nantes Atlantique, Université de Nantes

Un modèle multidimensionnel pour un processus d'analyse en ligne de résumés flous

Jury

Président	:	_____	_____
Rapporteurs	:	Claude CHRISMENT, Professeur	IRIT, Univ. Paul Sabatier (Toulouse III)
		Nacer BOUDJLIDA, Professeur	LORIA, Univ. Henri Poincaré (Nancy 1)
Examinateurs	:	Florence SÉDES, Professeur	IRIT, Univ. Paul Sabatier (Toulouse III)
		Noureddine MOUADDIB, Professeur	LINA, Univ. de Nantes
		Guillaume RASCHIA, Maître de conférences	LINA, Univ. de Nantes

Directeur de thèse : Noureddine MOUADDIB
Encadrant de thèse : Guillaume RASCHIA
Laboratoire : LABORATOIRE D'INFORMATIQUE DE NANTES ATLANTIQUE. 2, rue de la Houssinière, F-44322 NANTES CEDEX 3

N° ED 0366-XXX

UN MODÈLE MULTIDIMENSIONNEL POUR UN PROCESSUS D'ANALYSE EN LIGNE DE RÉSUMÉS FLOUS

A multidimensional model for on-line analytical processing of fuzzy summaries

Lamiaa Naoum

⋈

favet neptunus eunti

Université de Nantes

Lamiaa NAOUM

Un modèle multidimensionnel pour un processus d'analyse en ligne de résumés flous

xviii+176 p.

Ce document a été préparé avec LaTeX2$_\varepsilon$ et la classe `these-IRIN` version 0.92 de l'association de jeunes chercheurs en informatique LOGIN, Université de Nantes. La classe `these-IRIN` est disponible à l'adresse :

`http://www.sciences.univ-nantes.fr/info/Login/`

Impression : Naoum-these.tex – 21/11/2006 – 16:31

Révision pour la classe : $Id: these-IRIN.cls,v 1.3 2000/11/19 18:30:42 fred Exp

Á

Résumé

Le travail présenté dans cette thèse traite de l'exploration et de la manipulation des résumés de bases de données de taille significative. Les résumés produits par le système SaintEtiQ sont des vues matérialisées multi-niveaux de classes homogènes de données, présentées sous forme de collections d'étiquettes floues disponibles sur chaque attribut. La contribution de cette thèse repose sur trois points. En premier lieu nous avons défini un modèle de données logique appelé *partition de résumés*, par analogie avec les cubes de données OLAP, dans le but d'offrir à l'utilisateur final un outil de présentation des données sous forme condensée et adaptée à l'analyse. En second lieu, nous avons défini une collection d'opérateurs algébriques sur l'espace multidimensionnel des partitions de résumés. Ces opérateurs sont à la base d'une algèbre de manipulation des résumés. Cette algèbre prend en compte les spécificités du modèle de résumé que nous traitons. Nous avons adapté la majorité des opérateurs d'analyse proposés dans les systèmes OLAP. Ainsi, nous avons identifié : les opérateurs de base issus de l'algèbre relationnelle, les opérateurs de changement de granularité et les opérateurs de restructuration. Ces résultats offrent de nouvelles perspectives pour l'exploitation effective des résumés dans un système décisionnel. Finalement, pour compléter ce travail, nous nous sommes intéressés à la représentation des résumés et des partitions de résumés linguistiques, notamment pour en fournir une présentation claire et concise à l'utilisateur final. Appliquée à une hiérarchie de résumés produite par le système SaintEtiQ, l'approche tente de construire des prototypes flous représentant les résumés.

Mots-clés : Résumés de bases de données, cubes OLAP, concepts multidimensionnels flous, aide à la décision, prototypes flous.

Abstract

The work presented in this thesis deals with the subject of exploration and manipulation of database summaries with significant size. The summaries produced by SaintEtiQ system are multilevel materialized views of homogeneous data clusters, presented with a collections of fuzzy labels available on each attribute. Our thesis contribution is based on three points. Initially we defined a logical data model called *summaries partition*, by analogy with OLAP datacubes, with the aim of offering to the end-user a tool for data presentation in condensed form and adapted to the analysis. Secondly, we defined a collection of algebraic operators on the multidimensional space of summaries partitions. These operators are the base for an algebra for handling summaries. This algebra takes into account specificities of the summary model we deal with. We adapted the majority of the operators of analysis proposed in OLAP systems. Thus, we identified: core operators resulting from the relational algebra, operators of changing granularity and operators of reorganization. These results offer new prospects for the effective summaries exploitation in a decisional system. Finally, to complet this work, we were interested in the summaries and partitions representation, in particular to provide a clear and concise presentation of it to the end-user. Applied to a summaries hierarchy produced by the SaintEtiQ system, the approach tries to construct fuzzy prototypes representing the summaries.

Keywords: Databases summarization, OLAP datacubes, multidimensional vague concepts, decision support, fuzzy prototypes.

Remerciements

Je remercie...

Sommaire

Chapitre 1

Introduction générale

Problématique

Face à la mondialisation, les entreprises et plus globalement les organisations, se trouvent confrontées à des environnements de plus en plus complexes et compétitifs dans lesquels le pilotage et la prise de décision impliquent des choix qui doivent être faits dans des temps très courts tout en prenant en compte un volume d'information toujours plus important, la prise de décision dans les entreprises devient donc une préoccupation de première importance. Dans le but de pouvoir prendre des décisions pertinentes, les systèmes d'information dont sont équipées les entreprises nécessitent de nombreuses informations et donc utilisent souvent des bases de données volumineuses qui permettent à l'utilisateur d'avoir une vision complète afin de l'aider dans la prise de décision. Or, plus les bases de données sont volumineuses plus il devient difficile d'en extraire une information utile. De nombreux travaux se sont intéressés à ce problème de grand volume de données. Les méthodes d'analyse statistique telles que l'analyse factorielle, la classification hiérarchique et la régression peuvent ainsi être employées pour mettre en évidence les causalités et faire ressortir des ensembles logiques. Ainsi dans la thématique de traitement des données massives, la communauté de recherche en informatique s'est intéressée à la massification des données dans ses différents aspects : acquisition, stockage, transmission, traitement, modélisation, représentation, structuration, indexation, interrogation, comparaison, manipulation, classification, fusion, extraction de sens, apprentissage et visualisation.

Dès lors, afin d'accéder et d'exploiter, de manière décentralisée et en temps réel un grand volume de données de l'organisation, l'architecture des systèmes d'information s'est élargie aux systèmes décisionnels. Cette nouvelle génération de systèmes décisionnels aide les experts et les analystes en leur construisant des entrepôts avec des données déjà agrégées et destinées à l'étude d'un sujet particulier. Ces systèmes proposent des fonctionnalités d'extraction et d'analyse. Ils permettent notamment de collecter des informations provenant de sources différentes, d'exploiter ces données aux travers d'interfaces et d'opérateurs de manipulation et de représentation. L'importance des volumes de données mis en jeu dans ces systèmes décisionnels nécessite des mécanismes d'agrégation pour synthétiser l'information. Pour répondre à ces besoins, le traitement analytique en ligne (OLAP [1]) des systèmes décisionnels fournit une analyse s'appuyant sur un mécanisme d'interrogation interactive des données multidimensionnelles basé sur un ensemble d'opérateurs de navigation.

[1] On-Line Analytical Processing.

D'autres travaux se sont intéressés aux grands volumes de donnés en étudiant les résumés de données. L'objectif des résumés est de réduire le volume des données ainsi que de produire une connaissance à un niveau d'abstraction supérieur à celui des données d'origine. Plus particulièrement les résumés flous prennent en considération le bruit existant dans les grands volumes de données, et sont représentés dans le vocabulaire d'un utilisateur non expert comme pourrait l'être un décideur. Un tel résumé est donné par exemple par la phrase "la plupart des ventes sont faibles". Ils ont un intérêt certain en découverte de connaissance puisque les données sont alors résumées en termes linguistiques plus naturels pour les analystes.

Par ailleurs, l'exploration des données massives pour l'analyse est considérée comme un élément majeur d'un processus d'aide à la décision. En effet, l'extraction d'information enfouie au sein des données de l'entreprise aide à obtenir une vue homogène et fiable des données et, par la suite, facilite la prise de décision. Il est beaucoup plus simple de trouver une information pertinente dans une structure organisée pour la recherche de connaissance.

Contribution

Nos travaux de recherche s'inscrivent dans le cadre du laboratoire LINA (Laboratoire Informatique de Nantes Atlantique) au sein de l'équipe Atlas-Grim. Notre équipe a développé un système basé sur les concepts flous pour résumer les bases de données relationnelles. Ce système appelé SaintEtiQ a été développé dans le but de répondre à la problématique de massification de données. L'enjeu des résumés de données est la synthèse de l'information dans un but de fournir une représentation concise d'un grand volume de données.

Le système SaintEtiQ génère un grand volume de résumés nécessitant l'existence d'un mécanisme d'exploration à vocation analytique. Nos travaux de recherche visent à représenter et à manipuler les résumés flous des données qui sont générés par le système SaintEtiQ. Nous avons constaté qu'il devient nécessaire de trouver un modèle pour une présentation logique et claire d'une structure complexe contenant des résumés flous, afin de faciliter la manipulation d'information pour l'utilisateur.

La démarche de résumé en ligne s'inscrit bien dans une volonté de créer un parallèle avec les méthodes d'analyse en ligne que proposent les systèmes OLAP. Dans ce but il apparaît intéressant de proposer la mise au point d'une algèbre de manipulation des résumés qui serait le pendant de celle de manipulation des cubes de données.

L'objectif principal de ce travail est la proposition d'un modèle d'analyse en ligne pour les résumés flous. Pour ce faire, nous proposons une progression selon deux points :

- le premier consiste à réaliser un état de l'art. Cette étude s'intéresse aux systèmes décisionnels et précisément leurs moteurs d'analyse en ligne, ainsi qu'aux différentes méthodes qui traitent la problématique de la compression sémantique des données massives.
- le second permet d'apporter une contribution au système SaintEtiQ par la

proposition d'un modèle d'analyse en ligne des résumés flous, afin d'adapter le système à un processus d'aide à la décision.

Le modèle de données défini sur les résumés est un modèle multidimensionnel qui peut supporter l'application des opérateurs de haut niveau tels que les opérateurs d'analyse en ligne que nous retrouvons dans les moteurs OLAP. Sur la base de ce modèle, la définition d'une algèbre offre aux décideurs et à l'utilisateur final un outil convivial pour une interactivité lors de la manipulation et l'interrogation des résumés, dans le but d'extraire une information pertinente qui peut faciliter dans la prise de décision. Le dernier point de cette contribution est la proposition d'une méthode de présentation de résumés à l'utilisateur en lui garantissant une information synthétique et intelligible.

Organisation du document

La première partie de cette thèse décrit un état de l'art sur les travaux proposés dans la littérature concernant la prise de décision à partir des données volumineuses dans les systèmes décisionnels. Cette première partie présente également une étude bibliographique sur les travaux ayant été proposés dans le cadre de la compression sémantique des données, parmi lesquels on trouve le système SaintEtiQ, une approche basée sur les résumés linguistiques flous pour synthétiser des bases de données relationnelles de grand volume. Nous présentons aussi dans cette partie le système SaintEtiQ dans sa globalité. La description des éléments de base du système SaintEtiQ servira par la suite pour la compréhension de notre proposition.

Dans la deuxième partie de ce manuscrit, nous présentons notre contribution qui consiste à proposer un modèle qui supporte un processus d'analyse en ligne des résumés générés par SaintEtiQ. Cette proposition est présentée dans les chapitres 4, 5 et 6. Le chapitre 4 décrit notre modèle multidimensionnel basé sur les résumés flous et très proche d'un modèle décisionnel. La définition d'une algèbre qui propose différents opérateurs pour manipuler ces résumés est présentée dans le chapitre 5. Ce dernier présente un ensemble d'opérateurs semblables à ceux d'un processus OLAP ainsi que les propriétés de chacun. Le chapitre 6 propose de construire des prototypes flous à partir des résumés pour les présenter de façon intuitive à l'utilisateur.

Partie I

État de l'art

Introduction

Le volume de données disponible de nos jours dans les bases de données et dans les systèmes d'information des entreprises, ne cesse d'augmenter. Face à cette problématique de la gestion des données massives, plusieurs modèles ont été proposés. Nous étudions dans cette première partie du document l'état de l'art des systèmes dédiés à la gestion des grandes masses de données. Nous entendons par *"gestion"*: la modélisation, le stockage et la manipulation des données ainsi que leurs présentations sous une forme plus réduite.

Dans le premier chapitre, nous étudions les systèmes d'information décisionnels. Ces systèmes sont dédiés aux analystes et jouent un rôle de plus en plus important dans un processus d'aide à la décision. Nous allons étudier en détail l'architecture de ces systèmes ainsi que leurs caractéristiques. Nous nous intéressons aussi dans ce chapitre à l'état de l'art sur la manipulation à vocation analytique des données dans les systèmes décisionnels.

Le second chapitre est consacré à un tour d'horizon des principales méthodes proposées dans la littérature pour la réduction des grands volumes de données. Nous ferons une brève présentation à la fin de ce deuxième chapitre d'un processus nommé SaintEtiQ, qui génère des résumés à partir d'une base de données volumineuse.

L'objectif de cette première partie est d'étudier les éléments essentiels pour proposer notre contribution pour les résumés de SaintEtiQ. Nous allons montrer l'intérêt de l'analyse exploratoire et la manipulation des données dans les systèmes décisionnels qui vont nous inspirer pour enrichir le système SaintEtiQ.

Chapitre 2

Les systèmes d'information décisionnels

Si nous pouvions savoir où nous en sommes et vers quoi nous nous dirigeons, nous serions plus à même de juger quoi faire et comment faire.

— Abraham Lincoln, 1858.

2.1 Introduction

La Business Intelligence (BI), également *intelligence d'affaires* ou *informatique décisionnelle*, est apparue à la fin des années 70 avec les premiers infocentres. Des systèmes envoyaient des requêtes directement sur les serveurs de production, ce qui se révélait plutôt dangereux pour ces derniers. Dans les années 80, l'arrivée des bases de données relationnelles et du mode client/serveur a permis d'isoler l'informatique de production des dispositifs décisionnels. Dans la foulée, des acteurs spécialisés se sont lancés dans la définition de couches d'analyse métier, dans le but de masquer la complexité des structures de données.

La notion de *BI* englobe les solutions informatiques dont le but est de consolider les informations disponibles au sein des bases de données de l'entreprise. En effet, les entreprises, et plus globalement les organisations, se trouvent confrontées à des environnements de plus en plus complexes et compétitifs dans lesquels le pilotage implique des choix qui doivent être faits dans des temps très courts tout en prenant en compte un volume d'informations toujours plus important.

Dans ce chapitre nous présentons un état de l'art sur les systèmes décisionnels. Nous allons détailler dans la première section l'architecture générale d'un système décisionnel. La deuxième section est consacrée à la modélisation multidimensionnelle et la troisième section à l'exploitation des données dans les modèles multidimensionnels.

2.1.1 Définitions

Il existe deux grandes familles de Systèmes d'Information (SI). On distingue en effet :

– Les SI opérationnels et,
– Les SI décisionnels.

Les premiers, les SI opérationnels, sont utilisés pour la gestion du quotidien. Souvent, ils sont associés à des progiciels ou des applications développées pour répondre à une problématique métier. Leur objectif principal est la saisie puis les traitements de données, ainsi que la production de résultats en sortie. D'une manière générale, ces systèmes brassent un grand volume de données tout en garantissant un accès rapide à l'information. La réponse à des requêtes généralement peu complexes permet des temps de réponse relativement réduits.

Les seconds, les systèmes d'information décisionnels ont été définis dans [101] comme suit :

Definition 2.1 (Système d'Information Décisionnel)**.** *Un système d'information décisionnel (SID) est un ensemble de données organisées de façon spécifique, facilement accessible et appropriées à la prise de décision ou encore une représentation intelligente de ces données au travers d'outils spécialisés. La finalité d'un système décisionnel est le pilotage de l'entreprise.*

Les moyens de parvenir à une activité de pilotage passent par une information riche, pertinente, détaillée, historisée, fiable et stable. Le pilotage induit également l'utilisation d'outils d'analyse et de restitution puissants et adaptés à chacun des métiers concernés, ainsi qu'une forte capacité à faire évoluer les données et les outils.

Selon [31], dans son ouvrage *Construction du datawarehouse, du datamart et du dataweb*, les systèmes décisionnels constituent une synthèse d'informations opérationnelles, internes ou externes, choisies pour leur pertinence et leur transversalité fonctionnelles, et sont basés sur des structures particulières de stockage volumineux. Le principal intérêt d'un système décisionnel est d'offrir au décideur une vision transversale de l'entreprise intégrant toutes ses dimensions.

2.1.2 Objectifs

L'explosion des volumes de données et l'hétérogénéité des systèmes d'information, tels sont les nouveaux challenges auxquels font face les entreprises dans la conduite de leurs activités. À ces enjeux, s'ajoute aujourd'hui un contexte économique difficile imposant aux entreprises une exigence d'efficacité dans leurs décisions. Les éditeurs d'outils décisionnels, à travers leurs offres respectives, tiennent à se placer en première ligne afin de leur apporter des réponses. On peut parler de processus décisionnel lorsque les données de production sont valorisées en information. Cette valorisation est effective dès que l'on sort du monde de la production. Pour transformer des données en information, les systèmes décisionnels se fondent sur le rapprochement de données provenant de divers systèmes d'information internes ou externes, et sur la

synchronisation des différents flux d'informations. Le marché des applications décisionnelles rassemble plusieurs types d'outils pouvant être classés en trois catégories : les outils de constitution, les outils d'administration des bases de données, et les outils de restitution. Nous présentons plus en détails les différentes étapes qui constituent un système décisionnel dans la section suivante.

2.2 Architecture d'un système décisionnel

L'architecture type d'un système décisionnel peut être représentée en quatre niveaux. La figure 2.1 tirée de [101], nous montre cette architecture.

1. Le premier niveau est celui des sources de données, celui des systèmes *source*, à savoir le système de gestion de l'entreprise contenant les bases de données opérationnelles ainsi que des sources externes.
2. Le deuxième niveau concerne la récupération, la transformation des données, puis l'alimentation d'un entrepôt de données.
3. Le troisième niveau est celui de l'exploitation de l'entrepôt en question, celui-ci étant organisé de telle manière que l'on puisse, par fonction de l'entreprise, récupérer l'information mais avec une capacité d'agrégation des données par métier. Les *data marts*, considérés comme des bases de données métier, sont en fait extraits des entrepôts de données.
4. Le quatrième et dernier niveau fournit à l'utilisateur final le moyen de composer sa propre analyse et restitution des données.

Dans la suite de cette section nous détaillons chacun de ces quatre niveaux.

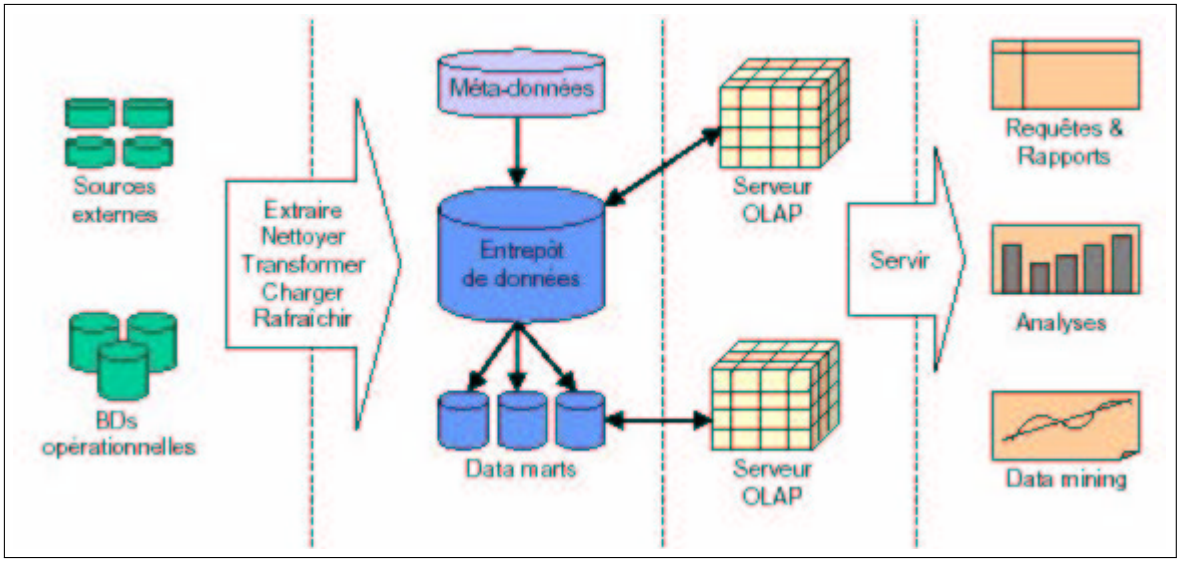


Figure 2.1 – Architecture d'un système décisionnel (tiré de [101])

2.2.1 Sources de données

La complexité des systèmes d'information se traduit par une multitude de lieux et de formats de stockage des données. Pour que des données soient exploitables, il est nécessaire de les agréger et de les nettoyer de tous les éléments non indispensables. Cette opération d'extraction et d'homogénéisation des données est assurée par la technologie *ETL (Extraction, Transformation and Loading)*.

Connecté aux différentes applications et bases de données, l'outil d'*ETL* se charge de récupérer ces données et de les centraliser dans une base de données particulière, l'entrepôt de données. Pour ce faire, le processus d'*ETL* respecte les trois étapes d'extraction, de transformation et de chargement.

La phase d'extraction consiste en l'identification et l'épuration des données, seules les données destinées à l'exploitation pour analyser un sujet bien précis seront gardées. Pour extraire les données utiles, l'outil d'*ETL* doit pouvoir se connecter aux différentes sources à disposition, qu'il s'agisse d'applications ou des bases en production. En conséquence, les *ETL* utilisent des moteurs d'extraction ou des programmes ad hoc générés par des outils dédiés. En ce sens, ils jouent un rôle d'intégration au niveau des données.

La phase de transformation regroupe les opérations de mise au format nécessaire des données, de calcul des données secondaires et de fusion ou d'éclatement des informations composites (par exemple le produit d'une quantité vendue et du prix peut devenir un total de vente). Cette étape de transformation comprend aussi une phase d'agrégation des données. Le niveau d'agrégation est choisi au moment de la construction de l'entrepôt et les données initiales seront perdues. C'est effectivement ce qui permet d'avoir des temps de réponse très courts.

Enfin, la phase de chargement a pour rôle de stocker les informations dans les entrepôts de données. Ce stockage dépend de la manière dont est administré l'entrepôt de données, le chargement de nouvelles données peut bien parfois écraser les données déjà existantes.

L'utilisation des *ETL* permet de filtrer et de nettoyer les données qui sont issues souvent de sources variées. La plupart des bases de données volumineuses souffrent de l'existence de données "bruitées". Ce "bruit" est dû à l'hétérogénéité et l'incohérence des données ainsi qu'à la présence de données manquantes ou encore à la duplication de certaines données. D'autres approches se sont intéressées à la réduction de ce "bruit"' existant dans un système d'information. Parmi ces approches, nous verrons dans la section 3.3.1 une façon de nettoyer les données primaires d'une base de données en utilisant des techniques de la théorie des sous-ensembles flous.

2.2.2 Entrepôts et magasins de données

2.2.2.1 Entrepôts de données

Le concept d'*entrepôt de données* a été proposé par W. H. Inmon en 1990 dans [42] pour répondre à des besoins d'analyse pour les décideurs que les systèmes transactionnels ne pouvaient pas fournir. Ralph Kimball dans [51] propose la définition

suivante d'un entrepôt de données:

Definition 2.2 (Entrepôt de données)**.** *Un entrepôt de données est un espace de stockage centralisé sur lequel repose un système décisionnel, son rôle est d'intégrer et de stocker l'information utile aux décideurs et de conserver l'historique des données pour supporter les analyses effectuées lors de la prise de décision.*

Plus précisément, dans l'ouvrage *"Le Data Warehouse"* de J. M. Franco [26], un entrepôt de données est une collection de données intégrées, thématiques, non volatiles et historisées pour la prise de décisions.

- *Des données intégrées.* Un entrepôt de données concerne les différents services et métiers de l'entreprise. Avant d'être intégrées dans l'entrepôt, et dans un souci de cohérence, les données doivent être mises en forme et unifiées. L'intégration nécessite une forte normalisation, une bonne gestion des référentiels et de la cohérence, une parfaite maîtrise de la sémantique et des règles de gestion s'appliquant aux données manipulées.
- *Des données thématiques.* Les données correspondent à des éléments d'analyse représentatifs des besoins utilisateurs. Elles constituent déjà un résultat d'analyse et une synthèse de l'information contenue dans le système décisionnel, et doivent être facilement accessibles et compréhensibles.
- *Des données non volatiles.* Afin de conserver la traçabilité des informations et des décisions prises, les informations stockées au sein de l'entrepôt de données ne peuvent pas être supprimées. Une requête lancée à différentes dates sur les mêmes données doit toujours retourner les mêmes résultats. Une donnée introduite dans l'entrepôt ne pourra donc plus être supprimée ni même modifiée. C'est pourquoi les données ne sont pas volatiles.
- *Des données historisées.* Chaque nouvelle insertion de données en provenance du système de production ne détruit pas les anciennes valeurs, mais crée une nouvelle occurrence de la donnée. L'historisation est nécessaire pour suivre dans le temps l'évolution des différentes valeurs des indicateurs à analyser. Ainsi, un référentiel temps doit être associé aux données afin de permettre l'identification dans la durée de valeurs précises.

Pour gérer physiquement et sémantiquement l'ensemble des données, l'entrepôt de données doit nécessairement disposer de *"données sur les données"*, à savoir de méta-données.

Les méta-données. Une donnée étant forcément liée à d'autres objets du système d'information, il est nécessaire de représenter, décrire et stocker les interactions avec d'autres données. Chaque donnée d'un système décisionnel doit alors être recensée avec précision. Il s'agit de connaître, pour chacune des données du système, sa provenance, sa signification, les transformations qu'elle doit subir avant d'intégrer le système décisionnel. En particulier, les méta-données doivent permettre de répondre aux questions suivantes : comment extraire une donnée, avec quelle périodicité, quelles transformations effectuer, quelle signification lui associer, les méta-données doivent également spécifier les droits d'accès associés à cette donnée. Elles ont également

en charge le stockage d'informations de nature hétérogène sur les données : date (de création, de modification), texte (description détaillée, règles de gestion, utilité), schéma (provenance, base).

Les données se trouvant dans les entrepôts de données sont souvent stockées sous forme de vues matérialisées. L'extraction des données à partir des entrepôts repose ainsi essentiellement sur la technique de ces vues matérialisées [33, 100].

Vues matérialisées. Une vue matérialisée est une table contenant les résultats d'une requête. Les vues améliorent l'exécution des requêtes en précalculant les opérations les plus coûteuses comme la jointure et l'agrégation, et en stockant leurs résultats dans la base. En conséquence, cetaines requêtes nécessitent seulement l'accès aux vues matérialisées [96] et sont exécutées plus rapidement. Une vue matérialisée consiste à calculer une vue exprimée sur une source de données et à stocker physiquement les données obtenues dans l'entrepôt. Les entrepôts de données sont souvent définis comme des vues matérialisées de données historisées stockées à des fins d'analyse. Ils doivent apporter une solution à un certain nombre de problèmes, liés notamment à la mise à jour efficace des données face aux flux constants de données hétérogènes produites dans les systèmes transactionnels, et liés également à la mise en œuvre d'applications décisionnelles.

2.2.2.2 Magasins de données

Les entrepôts de données sont organisés autour des sujets majeurs et des métiers de l'entreprise. Les données sont organisées par thème. L'intérêt de cette organisation réside dans le fait qu'il devient possible de réaliser des analyses sur des sujets transversaux aux structures fonctionnelles et organisationnelles de l'entreprise. Cette orientation permet également de faire des analyses par itération, sujet après sujet. L'intégration dans une structure unique s'avère indispensable pour éviter aux données concernées par plusieurs thèmes d'être dupliquées. Cependant dans la pratique, il existe également ce qu'on appelle des *data marts* ou *magasins de données*, c'est-à-dire des sous-ensembles d'un entrepôt de données, contenant des informations se rapportant à un secteur d'activité particulier de l'entreprise ou à un métier qui y est exercé (commercial, marketing, comptabilité, etc.). Dans [95], les magasins de données ont été définis comme suit :

Definition 2.3 (Magasin de Données)**.** *Un magasin de données est un extrait de l'entrepôt des données. Les données extraites sont adaptées à une classe de décideurs ou à un usage particulier (recherche de corrélation, logiciel de statistiques, . . .). L'organisation des données suit un modèle spécifique qui facilite les traitements décisionnels.*

Les magasins de données peuvent être perçus comme des petits entrepôts constitués d'un ensemble de données correspondant à un sujet précis, rendant très rapide les temps de réponses aux requêtes.

2.2.3 Serveurs OLAP

Comme le montre la figure 2.1, le troisième pillier d'un système décisionnel est celui des serveurs OLAP [1]. Dans cette section, nous proposons une brève présentation de la notion d'analyse en ligne et de serveur OLAP.

Le terme OLAP date de 1993. Il fut introduit par E. F. Codd [19], le père des bases de données relationnelles. Dans cet ouvrage [19], il définit un cahier des charges comprenant douze règles que doivent satisfaire les modèles OLAP. Les notions de dimensions et d'aggrégation ainsi que les concepts de base des systèmes OLAP seront détaillées plus loin dans ce chapitre. Ces règles sont :

- la multidimensionnalité : le modèle OLAP est multidimensionnel par nature,
- la transparence : l'emplacement physique du serveur OLAP est transparent pour l'utilisateur,
- l'accessibilité : l'utilisateur OLAP dispose de l'accessibilité à toutes les données nécessaires à ses analyses,
- la stabilité : la performance des rapports reste stable indépendamment du nombre de dimensions,
- architecture client-serveur : le serveur OLAP s'intègre dans une architecture client serveur,
- le dimensionnement : le dimensionnement est générique afin de ne pas fausser les analyses,
- la gestion complète : le serveur OLAP assure la gestion des données clairsemées,
- les multi-utilisateurs : le serveur OLAP offre un support multi-utilisateurs (gestion des mises à jour, intégrité, sécurité),
- l'inter-dimension : le serveur OLAP permet la réalisation d'opérations inter dimensions sans restriction,
- l'aspect intuitif : le serveur OLAP permet une manipulation intuitive des données,
- la flexibilité : la souplesse de l'édition des rapports est intrinsèque au modèle,
- l'analyse sans limites : le nombre de dimensions et de niveaux d'aggrégation possibles est suffisant pour autoriser les analyses les plus poussées.

Cette technologie OLAP a été proposée pour répondre aux limites que présentaient les systèmes transactionnels en matière d'analyse et de support pour la prise de décision.

OLTP versus OLAP

Les processus transactionnels en ligne OLTP [2], sont utilisés par les entreprises afin de gérer les informations contenues dans leurs systèmes opérationnels. Parmi les opérations typiques d'un système transactionnel, la mise à jour ponctuelle par des écrans prédéfinis, souvent répétitive, sur les données les plus récentes. Seulement ces systèmes OLTP ne peuvent répondre aux besoins spécifiques des entreprises pour

[1] On-Line Analytical Processing.
[2] On-Line Transactional Processing.

analyser l'information et supporter efficacement leurs processus d'aide à la décision.

Effectivement, contrairement aux applications transactionnelles OLTP, les applications OLAP sont réellement orientées utilisateur, dans leur choix et leur développement. Si les choix technologiques restent un enjeu important, le service rendu et l'efficacité apportée aux utilisateurs dans leur métier quotidien le sont tout autant.

Caractéristiques	**Processus OLTP**	**Processus OLAP**
Objectifs	Interrogation, Mise à jour	Analyse, Décision
Nature de données	Individuelles	Multidimensionnelles, agrégées, orientées utilisateur
Fraîcheur de données	Récentes, dynamique	Historiques, statiques
Traitement	Simple	Complexe, semi-automatique
Utilisateurs	Tout type	Décideurs
Temps de réponse	Rapide	Moins rapide

Table 2.1 – Comparaison des processus OLTP et OLAP

Le tableau 2.1 a été construit à partir de deux comparaisons des systèmes OLTP et OLAP issues de [55] et de [95]. Ce tableau résume les différences entre traitements OLAP et OLTP.

2.2.4 Les outils d'analyse

Suivant l'architecture d'un système décisionnel présentée à la figure 2.1, le dernier niveau est celui qui permet à l'utilisateur final, en l'occurence le décideur, d'exploiter les données stockées et de pouvoir les analyser et les restituer. C'est l'élément le plus important pour l'utilisateur car il correspond à la partie visible du système. Quelles que soient les solutions décisionnelles retenues, elles doivent être simples à utiliser et être compatibles avec les outils bureautiques existants. Nous présentons ici les trois types d'outils d'analyse (les requêtes, les rapports et les statistiques), souvent utilisés dans un système décisionnel. Cette partie de restitution de données sera étudiée d'une manière plus détaillée plus loin dans ce chapitre (voir section 2.4.1).

Les requêtes permettent à l'utilisateur d'accéder aux données et d'interroger les magasins ou les entrepôts de données selon ses besoins. Ces requêtes sont souvent exprimées à l'aide du langage SQL. Elles peuvent aussi utiliser des outils d'interrogation graphique fournis par la solution décisionnelle.

Les rapports quant à eux, sont souvent sous forme de tableaux de bord. Ces derniers sont construits à l'aide d'outils de visualisation et de navigation appelés EIS (Executive Information System). Ces outils sont souvent dotés d'une interface graphique très conviviale et très esthétique.

Les analyses de données dans un système décisionnel reposent souvent sur des outils d'analyse statistiques. D'ailleurs le concept d'OLAP, défini dans la section précédente, a beaucoup repris des bases de données statistiques (*BDS*). Dans ce

volet d'analyses statistiques, les systèmes décisionnels se basent sur les fonctionnalités qu'offrent les *BDS* pour la manipulation des données. Comme l'utilisation des modèles de graphes ou tabulaires. Dans [15], a été proposé le système *SAS* (*Statistical Analysis System*). Ce système considéré actuellement comme une solution décisionnelle, offre plusieurs fonctionnalités d'analyse dans le but de découvrire des connaissances. Pour ceci, des techniques statistiques comme l'échantillonage ainsi que des techniques de fouille de données sont utilisées dans la partie outils d'analyse de *SAS*.

Nous venons de détailler, dans cette section, les quatre grands modules qui constituent l'architecture d'un système décisionnel. Nous avons retenu deux notions importantes, les entrepôts et les magasins de données, qui constituent le noyau d'un système décisionnel. Une fois l'entrepôt construit, on construit des collections de données orientées sujet dans des magasins de données. Cette orientation sujet reflète la vision des analystes selon plusieurs axes (dimensions) d'analyse. Les données sont alors stockées selon ces axes, elles doivent correspondre à une structure adaptée à l'aspect multidimensionnel. La section suivante est consacrée à l'étude de la modélisation multidimensionnelle sur laquelle sont basés les systèmes d'information décisionnels.

2.3 Modélisation multidimensionnelle

Contrairement aux bases de données relationnelles, l'intérêt des bases de données décisionnelles ne se situe pas au niveau de l'individu (le n-uplet ou l'enregistrement) mais plutôt au niveau de l'identification des tendances dans un ensemble ou un groupe. L'objectif poursuivi est de permettre aux analystes d'avoir une vision des données qui supporte et aide leur processus de prise de décision. Afin de pouvoir analyser les données représentant l'activité d'une entreprise, il faut pouvoir les modéliser suivant des axes. Ainsi, à titre d'exemple, le chiffre d'affaire par catégorie de clients sur un produit donné se décline selon trois axes : montant de la facture, catégorie de clients et produit. De nombreux autres axes (ou dimensions) peuvent être définis, notamment en fonction de la zone géographique, du prix, ou de l'évolution dans le temps. D'après [95], la modélisation multidimensionnelle est définie comme suit :

Definition 2.4 (Modélisation multidimensionnelle)**.** *La modélisation multidimensionnelle consiste à considérer un sujet analysé comme un point dans un espace à plusieurs dimensions. Les données sont organisées de manière à mettre en évidence le sujet analysé et les différentes perspectives de l'analyse.*

Une base de données multidimensionnelle stocke alors les données de manière à permettre des analyses décisionnelles. À l'intersection de plusieurs dimensions se trouvent des valeurs, également nommées indicateurs ou variables. Elles peuvent par exemple quantifier le montant des ventes, ou le chiffre d'affaire. Ces valeurs sont extraites de la base de production, ou peuvent être calculées par le moteur OLAP. Dans ce cas, la valeur résulte d'une opération mathématique simple, telle qu'une

addition, mais elle peut nécessiter un traitement plus complexe, tel qu'un traitement statistique.

2.3.1 Modélisation conceptuelle

Conceptuellement, la modélisation multidimensionnelle a donné naissance à différents concepts dont les principaux sont les suivants :

- *Fait.* Un fait modélise le sujet d'analyse. Un fait est formé de *mesures* correspondant aux informations sur l'activité analysée.
- *Mesure.* Une mesure est la valeur qui associe un fait à un axe d'analyse (dimension).
- *Dimension.* Une dimension modélise l'objet d'analyse; elle se compose de paramètres qui peuvent faire varier les mesures.
- *Hiérarchie.* Une hiérarchie permet d'organiser les membres d'une dimension selon une relation *est plus fin* conformément à leur niveau de détail.

Dans les travaux de Ravat et al. [86], sur la modélisation et la fusion de données multidimensionnelles, les concepts de base de la modélisation multidimensionnelle ont été définis formellement comme ci-après :

Definition 2.5 (Fait)**.** *Un fait F_j est défini par $(N_{F_j}, M_{F_j}, I_{F_j})$ où*

- *N_{F_j} est le nom de fait,*
- *$M_{F_j} = \{m_1, m_2, \ldots, m_w\}$ est un ensemble de mesures (ou indicateurs d'analyse),*
- *$I_{F_j} = \{I_{F_1}, I_{F_2}, \ldots, I_{F_w}\}$ est l'ensemble des instances de F.*

Definition 2.6 (Dimension)**.** *Une dimension D_i est définie par $(N_{D_i}, A_{D_i}, H_{D_i}, I_{D_i})$ où*

- *N_{D_i} est le nom de la dimension,*
- *$A_{D_i} = \left\{a_{D_{i-1}}, a_{D_{i-2}}, \ldots, a_{D_{i-u}}\right\}$ est un ensemble d'attributs,*
- *$A_{D_i} = \left\{h_{D_{i-1}}, h_{D_{i-2}}, \ldots, h_{D_{i-y}}\right\}$ est un ensemble de hiérarchies,*
- *$I_{D_i} = \left\{I_{D_{i-1}}, I_{D_{i-2}}, \ldots\right\}$ est l'ensemble des instances de D_i.*

Definition 2.7 (Hiérarchie)**.** *Une hiérarchie représente une perspective d'analyse précisant les niveaux de granularité auxquels peuvent être manipulés les indicateurs d'analyse. Une hiérarchie $h_{D_{i_x}}$ définie sur la dimension D_i est un chemin élémentaire acyclique débutant sur l'attribut de plus faible granularité et se terminant par u attribut de plus forte granularité. Elle est définie par $(N_{D_{i-x}}, Parama_{D_{i-x}}, Suppl_{D_{i-x}})$ où*

- *$N_{D_{i-x}}$ est le nom de la hiérarchie,*
- *$Param_{D_{i-x}} = \langle a_{Di-k}, a_{Di-l}, \ldots, a_{Di-z}\rangle$ est un ensemble ordonné décrivant la hiérarchie des attributs (chaque attribut est appelé paramètre de la hiérarchie et correspond à un niveau de granularité d'analyse).*
- *$Suppl_{D_{i-x}} : Param_{D_i-x} \rightarrow 2^{A_{Di}-Param_{Di-x}}$ est une application spécifiant les attributs faibles qui complètent la sémantique des paramètres (chaque paramètre est associé à un ensemble d'attributs faibles).*

L'un des modèles les plus populaires des systèmes OLAP, est le cube de données. La notion du cube a été proposée par [32] en 1997. Elle a fait l'objet de nombreux travaux [32, 40, 66, 98]. Le cube de données est alors défini comme suit :

Definition 2.8 (Cube de Données)**.** *Un cube est un ensemble de données organisées selon des dimensions. On appelle mesure la valeur contenue dans une cellule du cube, associée aux valeurs prises sur les dimensions composant le cube.*

Exemple 2.1 (Exemple d'un cube de données).
Prenons l'exemple représenté par la figure 2.2, tiré de [90]. Il s'agit d'un cube à trois dimensions : le produit, les points de vente et la date. La mesure du cube peut être constituée par exemple des résultats des ventes des produits pour différentes villes à différents mois.

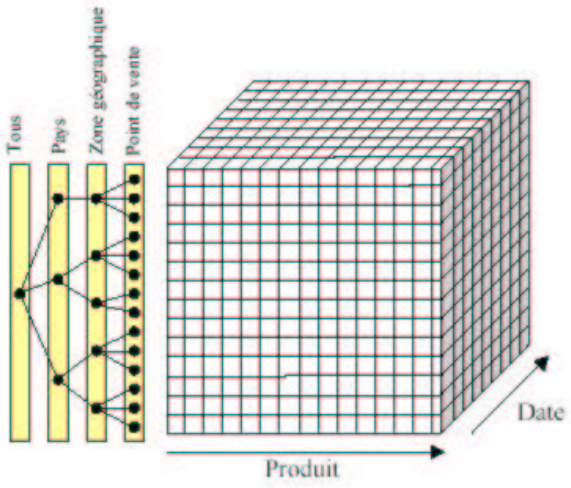


Figure 2.2 – Exemple de cube de données

2.3.2 Modélisation physique

Les concepts définis dans une structure multidimensionnelle ont été représentés par différentes techniques de modélisation multidimensionnelle physique. Plusieurs propositions de schémas existent dans la littérature [2, 34, 78, 18]. Les plus courants sont les trois schémas que nous présentons ci-dessous.

- Le modèle en *étoile* (star schema) : un schéma en étoile est constitué du fait central et des dimensions. Ce modèle représente de manière dénormalisée les dimensions. Ce schéma est défini formellement comme suit :
  Definition 2.9 (Schéma en étoile)**.** *Un schéma en étoile est noté* (F, D) *où:*
  – F *est un ensemble de faits ayant* m *mesure avec* $\{F.M_q, 1 \leq k \leq m\}$,

- $D = \{D_s, 1 \leq s \leq r\}$ *un ensemble de* r *dimension où chaque* D_s *contient un ensemble de* n_s *attributs* $D_s.A_i, 1 \leq i \leq n_s$.

– Le modèle en *flocon* (snowflake) : ce schéma consiste à décomposer les dimensions du modèle en étoile en sous hiérarchies. Le fait est conservé et les dimensions sont éclatées conformément à sa hiérarchie de paramètres. L'avantage de ce modèle c'est la formalisation d'une hiérarchie au sein d'une dimension. Cet avantage permet d'éviter le problème de redondance qu'on peut trouver dans le modèle en étoile. D'une manière plus formelle, le schéma en flocon a la définition suivante :

Definition 2.10 (Schéma en flocon)**.** *Un schéma en flocon est noté* (F, H) *où :*

- F *est un ensemble de faits ayant* m *mesure avec* $\{F.M_q, 1 \leq k \leq m\}$,
- $H = \{H_s, 1 \leq s \leq r\}$ *un ensemble de* r *hiérarchies indépendantes.*

– Le modèle en *constellation* : cette technique issue du modèle en étoile, et appelée modélisation en *constellation*, consiste à fusionner plusieurs modèles en étoile [62] qui utilisent des dimensions communes. Un tel modèle comprend donc plusieurs faits et des dimensions, celles ci pouvant être communes à plusieurs faits. Le schéma en constellation a la définition suivante :

Definition 2.11 (Schéma en constellation)**.** *Une constellation* C *est définie par*
$(N^C, F^C, D^C, Star^C)$ *où :*

- N^C *: nom de la constellation,*
- $F^C = \{F_1, F_2, \ldots, F_D\}$ *: ensemble de faits,*
- $D^C = \{D_1, D_2, \ldots, D_q\}$ *: ensemble de dimensions,*
- $Star^C : F^C \rightarrow 2^{D^C}$ *: fonction associant les faits aux dimensions.*

Afin de faciliter la lecture de ces trois modèles, nous les illustrons à l'aide d'une représentation graphique facilement compréhensible tirée de [95]. La figure 2.3 décrit le schéma en étoile modélisant les analyses des quantités et des montants de médicaments dans les pharmacies selon trois dimensions : le temps, la catégorie et la situation géographique. La figure 2.5 illustre la modélisation en constellation. Une constellation est constituée de deux schémas en étoile : l'un correspond aux ventes effectuées dans les pharmacies et l'autre analyse les prescriptions des médecins. La figure 2.4 illustre la modélisation en flocon, le même modèle en étoile de la figure 2.3 est décrit en dénormalisant chacune de ses dimensions, formant ainsi une sorte de flocon.

2.3.3 Modélisation logique

La modélisation logique multidimensionnelle est possible selon un modèle ROLAP ou un modèle MOLAP. Techniquement, le stockage physique des données est principalement fait selon ces deux modèles. D'autres modèles existent comme HOLAP[3] ou DOLAP[4] mais sont rarement utilisés. Nous présentons ici les modèles ROLAP et

[3]Hybrid OLAP : fusionne ROLAP et MOLAP.
[4]Desk OLAP : une version OLAP sur le poste client seulement.

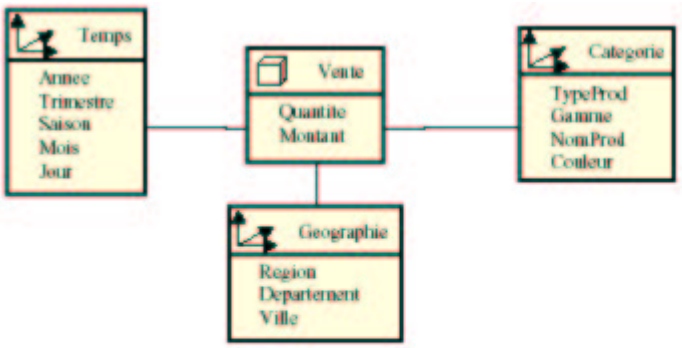


Figure 2.3 – Exemple d'une modélisation en étoile (tiré de [95])

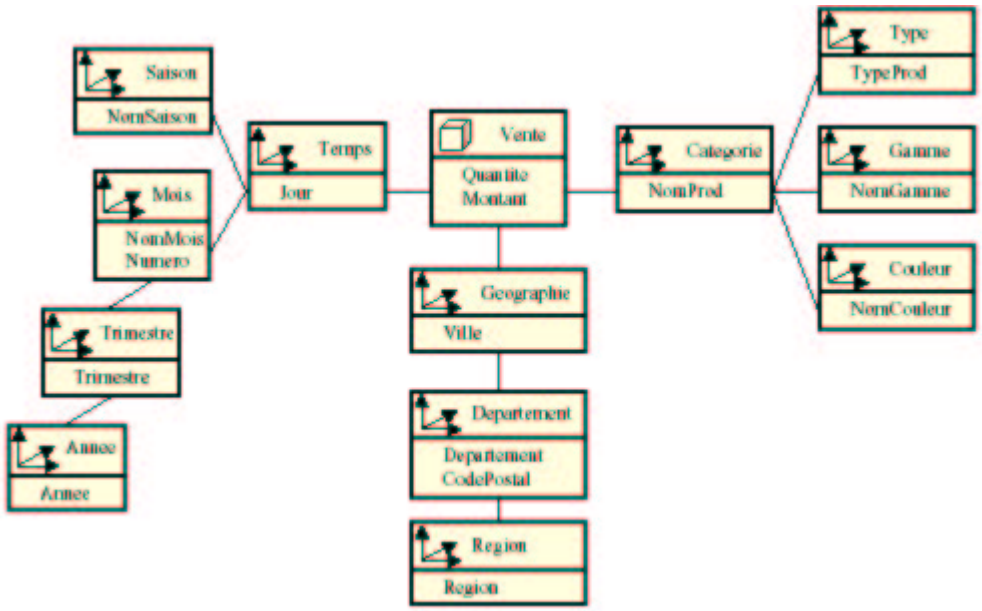


Figure 2.4 – Exemple d'une modélisation en flocon (tiré de [95])

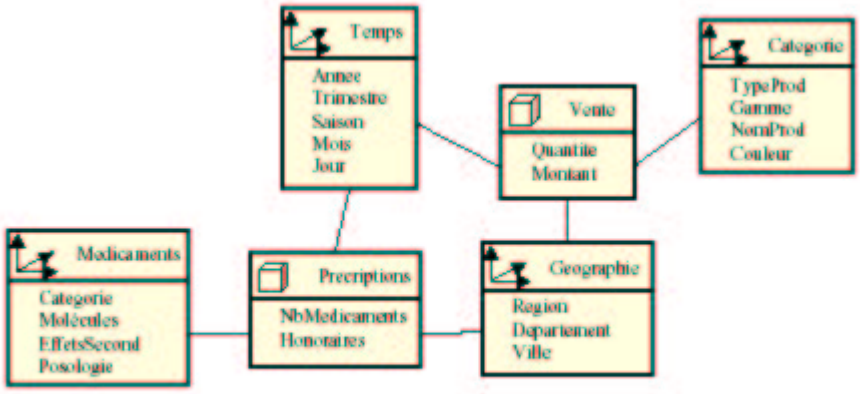


Figure 2.5 – Exemple d'une modélisation en constellation (tiré de [95])

MOLAP.

2.3.3.1 OLAP relationnel

ROLAP, pour Relational On Line Analytical Processing, stocke physiquement les données dans un Système de Gestion de Base de Données Relationnel (SGBDR). A chaque interrogation de l'utilisateur, la base relationnelle alimente dynamiquement un cube multidimensionnel virtuel. Dans l'approche ROLAP, le modèle multidimensionnel est traduit par des tables relationnelles. Ainsi chaque *fait* correspond à une *table de faits* et chaque *dimension* à une *table de dimensions*. Nous présentons ici quelques modèles ROLAP, proposés dans la littérature.

- **Le modèle de *GRAY et al.*** Ce modèle [32] est l'extension du modèle relationnel. L'opérateur *datacube* est une extension d'une table relationnelle, en calculant toutes les combinaisons possibles des attributs de la relation, et ensuite, en agrégeant ces différentes combinaisons. Dans ce modèle a été introduit l'opérateur *CUBE* qui calcule toutes les agrégations marginales des données connues à un niveau de précision détaillé.
- **Le modèle de *LI et WANG.*** Dans le modèle proposé par Li et Wang [63], les dimensions sont modélisées comme des relations appelées *relations de dimensions*, et les cubes par le produit cartésien de ces dimensions avec les mesures. Pour les hiérarchies, on considère des relations et des partitionnements de données selon les valeurs d'attributs.
- **Le modèle de *HAR.*** Ce modèle [40] propose de gérer les données sous une forme relationnelle et de n'utiliser une forme de cellules pour les présenter qu'en cas de nécessité. Le point prédominant de ce modèle est donc le choix des cellules à matérialiser, parce que seulement les parties nécessaires du cube sont matérialisées. Dans ce modèle, les données multidimensionnelles sont gérées

sous forme d'hypercubes construits le long de dimensions. Une dimension $\mathcal{L}$ est constituée d'un ensemble d'attributs partiellement ordonné. Les hiérarchies sont gérées par des treillis $(\mathcal{L}, \preceq)$.

- **Le modèle de *BAR.*** Dans ce modèle [7] une base de données multidimensionnelle est considérée comme un entrepôt relationnel, dédié aux décideurs et aux analystes, où l'information serait organisée en schéma en étoile. Cette base est composée d'un ensemble de tables $(\{D_1, \ldots, D_n\}, \mathcal{F})$, formant un schéma en étoile, où D_i est la table de dimensions avec une table des faits notée $\mathcal{F}$ celle reliant les tables de dimensions. Les hiérarchies sur les dimensions sont définies suivant les dépendances fonctionnelles existant sur les attributs des tables de dimensions.
- **Le modèle de *GYSSENS et al.*** Dans [34], les auteurs ont proposé un modèle tabulaire de base de données avec des tables multidimensionnelles. Les mesures sont considérées comme des attributs supplémentaires de la table des faits. La hiérarchie est stockée comme un attribut séparé des dimensions. En revanche, une dimension est décrite par un nom associé à une hiérarchie.

2.3.3.2 OLAP multidimensionnel

MOLAP, pour Multidimentional On Line Analytical Processing, stocke physiquement les données dans une base multidimensionnelle. Le cube est alors fourni directement. La structure multidimensionnelle utilisée dans les systèmes MOLAP, est un tableau à n dimensions. Nous trouvons également l'appellation d'*hypercube* ou de *cube* pour désigner cette structure. Dans ce qui suit, nous présentons quelques modèles MOLAP de la littérature, et la manière dont chaque modèle définit les concepts de base d'une structure multidimensionnelle (fait, mesure, ...).

- **Le modèle de *AGRAWAL et al.*** Ce modèle [2], introduit la notion d'un cube noté C et la défini par les composants suivants :
  - k dimensions, chaque dimension D_i est associée à un domaine de valeurs dom_i avec $(i = 1, \ldots, k)$,
  - les éléments $E(C)(d_1, \ldots, d_k)$, appartenant à un ensemble V des valeurs du cube, où $d_i \in dom_i$ pour $i = 1, \ldots, k$. Ces éléments peuvent être soit un n-uplet, soit 0 ou 1. Le cas de 0 signifie que la combinaison des valeurs de dimensions n'existe pas dans la base. Dans le cas où l'on a la valeur 1, alors il existe une combinaison, et dans le cas où l'élément est un n-uplet, alors on dispose d'une information complémentaire pour cette combinaison de valeurs de dimensions.

  Le modèle repose sur la notion de cube avec un ensemble d'opérations. Ces opérations sont closes, c'est à dire si on applique une opération sur un cube elle nous donne en résultat un cube aussi. Les mesures et les dimensions sont traitées de manière symétrique. Les sélections et les agrégations sont autorisées sur toutes les dimensions et les mesures.
- **Le modèle de *CABIBO et TORLONE.*** Cabibbo et Torlone dans [16] proposent un modèle fondé sur les notions de *dimension* et de *f-table*. Une *f-table*

stocke les données factuelles, elle est de la forme $f\left[A_1 : l_1\left\langle d_1\right\rangle, \ldots, A_n : l_n\left\langle d_n\right\rangle\right] : l_0\left\langle d_0\right\rangle$, où f est un nom d'un sujet à analyser, chaque $A_i (1 \leq i \leq n)$ est un attribut de f et chaque l_i est un niveau de la dimension d_i. Les dimensions sont organisées sous forme de hiérarchies où les niveaux correspondent aux différentes possibilités de granularité de données. Les hiérarchies sont représentées sous forme de treillis, avec des fonctions de généralisation définissant comment naviguer entre les différents niveaux de la hiérarchie.

– **Le modèle de *VASSILIADIS.*** Le modèle de Vassiliadis [99] représente les dimensions et les hiérarchies d'une manière explicite. Ce modèle est fondé sur le concept de *cube de base* qui représente les données au niveau le plus détaillé. Une hiérarchie est représentée sous forme d'un treillis et tous les autres cubes sont calculés à partir du cube de base.

2.3.3.3 MOLAP vs ROLAP

Chaque moteur de ROLAP ou de MOLAP possède ses avantages et ses inconvénients : MOLAP présente de bons temps de réponse dans le cas d'une information très structurée. En revanche ROLAP, plus tributaire des temps de réponse du SGBDR, propose une plus grande liberté d'analyse sur des volumes importants. La solution ROLAP bien que moins rapide pour accéder aux informations permet néanmoins un stockage plus important que la solution MOLAP. C'est pourquoi la définition de la structure de la base est très importante : elle doit être déterminée en fonction du type de requêtes qu'elle devra traiter, de la structure de l'information et de la volumétrie des données. Cette étape est importante car il est pratiquement impossible de modifier le principe de conception sans réécrire le système dans son entier.

Néanmoins, si l'utilisation des moteurs multidimensionnels reste relativement confidentielle, c'est en raison de certains inconvénients majeurs. En effet, il n'existe que des systèmes propriétaires, ayant leur propre architecture, leur propre langage d'accès aux données. De plus, ces moteurs nécessitent des machines puissantes pour pouvoir supporter les traitements d'agrégation et d'indexation, et des compétences techniques spécifiques pour les alimenter et les administrer. Tout ceci contribue à faire des systèmes décisionnels des systèmes coûteux. Par ailleurs, le modèle ne peut répondre à toutes les requêtes : seuls les résultats pré-agrégés lors de la constitution du cube peuvent être obtenus. Ajouter une dimension ou un indicateur au modèle nécessite la restructuration du système. Et, cette technologie atteint ses limites lorsque la quantité de données à intégrer dans le cube devient trop importante, le pré-calcul des indicateurs à l'intersection de tous les axes devenant matériellement impossible : la complexité du calcul est une fonction exponentielle de la quantité de données.

La question du choix entre base multidimensionnelle ou entrepôt de données reste ouverte et dépend évidemment du contexte et des besoins de l'utilisateur.

Nous avons choisi de citer ci-dessus les modèles les plus connus dans la littérature des travaux sur la modélisation multidimensionnelle. D'autres modèles existent comme le modèle de Datta [22], le modèle de Medelzoone et Vaisman [97], le modèle de Pourabbas [80], l'extension du modèle multidimensionnel à l'objet dans [95] et

aussi l'extension aux bases de données floues dans [55].

Indépendamment des modèles présentés, nous proposons de définir formellement un cube de données. Pour ceci nous concluons de cette section que les données multidimensionnelles sont modélisées sous la forme d'hypercubes dans lesquels les dimensions constituent des axes d'analyse indépendants, et les sujets d'analyse, ou faits sont caractérisés par des mesures qui sont pré-calculées à l'aide de fonctions d'agrégations selon les différentes granularités définies par le schéma hiérarchique de chaque dimension. Les mesures sont représentées dans les cellules d'un cube selon les différents niveaux de chaque dimension, ainsi de manière formelle, nous avons retenu la définition d'un cube tirée de [1] :

Definition 2.12 (Cube)**.** *Un cube c est une fonction injective d'un espace fini n-dimensionnel (défini par le produit cartésien de n niveaux indépendants* $\{L_1, \ldots, L_n\}$*, vers un ensemble d'instances de cellule* C_c *:*

$$c : L_1 \times \ldots \times L_n \rightarrow C_c, \quad injective$$

Nous nous intéressons dans la section suivante à l'exploitation des données multidimensionnelles. Bien que les opérateurs de manipulation des modèles cités dans ce chapitre ne soient pas encore uniformisés, nous allons recenser les différentes façons utilisées pour la manipulation, la visualisation et la restitution des données.

2.4 Exploitation des données multidimensionnelles

Parmi les aspects qui font l'importance des systèmes décisionnels, nous pouvons citer les deux points suivants :

- Les interfaces utilisateurs concernant la restitution et la visualisation des données.
- La manipulation des données grâce à la possibilité de navigation dans les données à différents niveaux d'abstraction.

Comme le souligne E. F. Codd [19], l'interface utilisateur dans un modèle multidimensionnel doit offrir une représentation synthétique et flexible, ainsi qu'une animation du modèle de données.

2.4.1 Restitution des données multidimensionnelles

2.4.1.1 Production de rapports et diffusion d'information

Diffusion de l'information. Jusqu'à l'apparition d'Internet, la diffusion d'information était très limitée. Et la diffusion sur support papier était parfois même privilégiée

pour des raisons de coûts et de simplicité. Mais avec la généralisation de l'Internet dans les organisations, de nouveaux moyens de diffusion de l'information ont été expérimentés. Une solution consiste à mettre à disposition les états via un serveur Web, Intranet ou Extranet. L'accès aux rapports devra être sécurisé en fonction des droits associés à l'utilisateur. Cette solution présente l'avantage de ne pas nécessiter l'installation d'un logiciel sur les postes des utilisateurs. En outre, les calculs peuvent être réalisés en temps réel, offrant ainsi aux rapports des données constamment à jour.

Production des rapports. Le terme utilisé pour la production des rapports dans les SID est le *reporting*. Les navigateurs constituent le premier outil de restitution des données et un moyen aisé d'accès à l'information. Les outils de reporting répondent au besoin principal des utilisateurs, à savoir produire des rapports et des tableaux de bords, tâche pour laquelle ils sont parfaitement adaptés. Fonctionnant sur la base d'un référentiel métier commun (*Univers* de *Business Object* par exemple) ils rendent transparents pour l'utilisateur les mécanismes informatiques inhérents à l'édition des rapports et la manipulation des données. Il est à noter que leur forme est définie à l'avance, leur diffusion étant assurée de respecter la forme initiale. Par contre, ces outils ne conviennent pas aux utilisateurs désireux de manipuler les données directement, ou souhaitant tout simplement utiliser leurs propres mises en page. Sur le marché des outils de reporting [5], on trouve le leader *Business Object* [6], suivi de *Cognos* [7], *Brio* [8], *Hummingbird* [9] et d'autres. Parallèlement à ces outils de reporting, on trouve des outils d'analyse, abordés ci-dessous.

2.4.1.2 Les fonctions d'analyse

Le monde décisionnel fait intervenir deux métiers, les décideurs et les analystes. Ces acteurs ont besoin d'indicateurs clés pour analyser les données et parfois prendre rapidement des décisions. Les systèmes décisionnels leurs proposent des outils d'analyse statistique ou de fouille de données afin de mieux exploiter d'importantes masses d'informations et de pouvoir en tirer un enseignement.

Analyse de données. Si certains acteurs de l'entreprise ont un besoin centré autour du reporting, d'autres ont besoin en revanche d'analyser de manière plus précise les données. Il s'agit d'expliquer les anomalies et leurs origines. Il s'agit également de mettre en lumière des phénomènes extrêmes dans la structure même d'un résultat chiffré : si un commercial fait par exemple un chiffre d'affaire record, et que parallèlement, un autre commercial obtient des résultats médiocres, le chiffre en sortie semblera normal, les deux résultats s'annulant réciproquement. Une analyse des données permet de révéler les disparités et d'expliquer des phénomènes apparemment normaux. Dans cette logique, ont été définies des opérations de spécialisation (per-

[5] une suite de solutions commerciales est présentée en annexe de ce document.
[6] http://www.france.businessobjects.com/
[7] http://www.cognos.com/
[8] http://www.brio.com/fr/
[9] http://www.hummingbird.com/fr

mettant de visualiser le détail composant une information) ou de généralisation (qui consiste à visualiser d'autres indicateurs pour expliquer une information [10]). Très interactifs, ces outils permettent de faire des simulations, de tester des scénarios. Ils englobent par ailleurs des fonctionnalités d'alerte : par exemple, un résultat négatif pourra être symbolisé par une police de couleur rouge, et une tendance à la hausse se matérialisera par une flèche montante.

Outils statistiques et de fouille de données. La fouille de données (data mining) est une technique d'analyse automatique pour relever des tendances ou des corrélations cachées parmi des masses de données, ou encore pour détecter des informations stratégiques ou découvrir de nouvelles connaissances, en s'appuyant sur des méthodes de traitement statistique. Pour ce faire, il se base sur des méthodes statistiques comme la corrélation ou les arbres de décision, et sur l'intelligence artificielle avec entre autres les réseaux de neurones. Par exemple, la fouille de données est souvent utilisée par les opérateurs pour prévenir le *churn*, c'est à dire le moment même où les abonnés d'un réseau de télécommunication vont devenir infidèles. Dans ce cadre, seule une historisation du comportement de l'utilisateur (achat de services, degré d'utilisation...) permet de définir des probabilités en fonction de critères tirés d'observations préalables.

Les requêteurs graphiques sont simples et facilitent l'accès aux informations, mais ils restent limités pour un processus d'analyse complexe. Les outils spécifiques OLAP offrent de nombreuses possibilités de manipulation multidimensionnelle mais il sont difficiles d'accès pour les non informaticiens. Enfin, les SGBD relationnels tendent à intégrer certains concepts de l'approche OLAP mais ils restent incomplets et ne profitent pas de la puissance des concepts multidimensionnels.

2.4.2 Visualisation des données multidimensionnelles

P. Marcel montre dans [68], que la modélisation sous forme de relations devient inappropriée pour supporter la prise de décision et les analyses multidimensionnelles de type OLAP. Les données multidimensionnelles sont visualisées sous une forme reflétant leur caractère multidimensionnel. Cela peut se faire en représentant les données dans une table selon deux dimensions (tableau 2.3) ou encore selon plusieurs dimensions (trois dimensions par exemple sur les figures 2.6 et 2.7).

Considérons un exemple parmi les plus récurrents dans la littérature OLAP [35, 68], pour ceci prenons la relation décrite dans le tableau 2.2 qui représente les quantités de pièces vendues en 2004 dans 4 régions différentes.

Visualisation tabulaire bidimensionnelle

Cette représentation peut se faire selon une table croisée à deux dimensions. On peut visualiser les données de la table 2.2 présentées selon deux dimensions on choisit ici d'afficher les quantités de ventes en fonction des pièces (en ligne) et de la région (en colonne).

On note qu'une table permet de représenter le produit cartésien des différentes

pièces	régions	ventes
écrous	ouest	60
écrous	est	50
écrous	sud	40
clous	nord	40
clous	est	70
vis	ouest	50
vis	sud	50
vis	nord	60

Table 2.2 – La relation ventes 2004

	ouest	est	nord	sud
écrous	60	50		40
clous		70	40	
vis	50		60	50

Table 2.3 – Exemple d'une représentation sous forme de table croisée

valeurs d'une relation. Or, toutes les combinaisons ne sont pas connues c'est pour cela qu'on doit représenter cette absence par une case vide, par exemple les ventes des clous dans la région ouest en 2004 (voir tableau 2.3). On note aussi que la représentation tabulaire est possible pour les données qu'elles soient stockées dans des relations de type ROLAP ou de type MOLAP.

Visualisation graphique multidimensionnelle

Cette représentation se fait en associant une dimension de l'espace à chaque attribut (colonne) de la table donnée. Dans ce cas la relation peut être décrite dans l'espace des trois dimensions (pièces, régions et ventes) chacune associée à un attribut (une colonne de la table). La figure 2.6 montre ce type de représentation pour la relation ventes du tableau 2.2.

Visualisation par cubes de données

Maintenant, si l'analyste cherche à représenter les données de plusieurs relations relatives aux ventes de différentes années. La représentation bidimensionnelle illustrée dans la table 2.3 peut être étendue à une représentation multidimensionnelle, sous forme de cube, comme sur la figure 2.7. Les données de la table 2.2 sont représentées comme celles de l'année 2004, le cube représente aussi les autres relations en prenant en compte la dimension du temps (années). Les dimensions du cube sont présentées par des axes. Chaque axe est gradué par des valeurs appelées paramètres ou membres. La valeur contenue dans chacune des cellules du cube est appelée *mesure* ou *élément*.

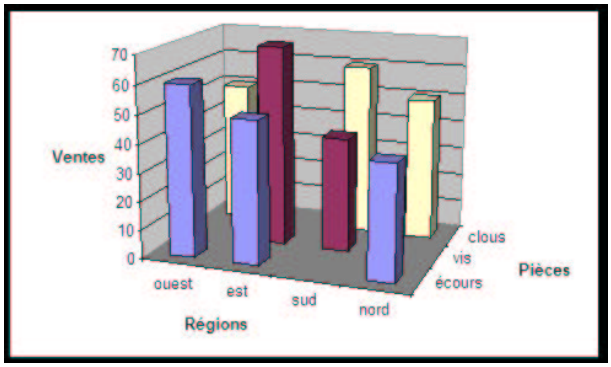


Figure 2.6 – Exemple d'une représentation sous forme de barres

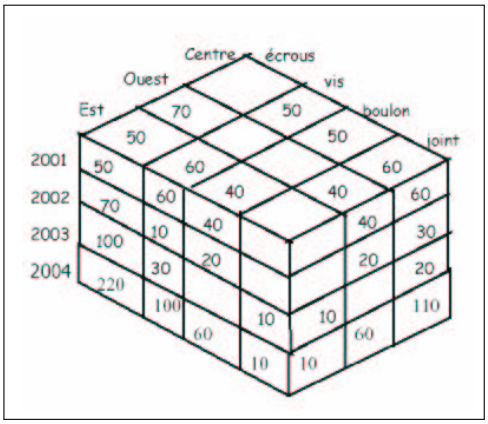


Figure 2.7 – Exemple d'une représentation sous forme de cube (tiré de [68])

La littérature sur la représentation et la visualisation des cubes est encore pauvre. Ce n'est que récemment que les études qui s'intéressent aux modèles multidimensionnels commencent à élargir le champ d'étude. Elles s'intéressent à des approches de représentation multidimensionnelle dédiées à la visualisation d'un cube à l'écran dans [69, 70] et de trouver une représentation attribuée aux tables multidimensionnelles et destinée à l'exploration dans [14, 17] ainsi qu'à l'optimisation des requêtes OLAP à un niveau logique [30].

Les travaux de [66] proposent un nouveau modèle pour présenter les cubes de données multidimensionelles appelé *CPM* pour *Cube Presentation Model*. Ce modèle de représentation est de type visualisation géométrique. Il se base sur la perception humaine des données du cube dans l'espace. Il est considéré comme une technique avancée pour la visusalisation des cubes OLAP. Par ailleurs, dans [39] a été proposé un outil appelé *CubeExplorer* qui, parmi ses multiples fonctionnalités d'analyse et de construction de cube, offre une interface graphique conviviale qui permet à l'utilisateur de spécifier de nouvelles contraintes, mesures ou autres paramètres pour les besoins d'analyse et de prise de décision.

D'autres travaux ont étudié la visualisation et l'exploration de grande collections de documents. L'approche proposée dans [71, 73] permet de caractériser le sous-ensemble de documents répondant aux centres d'intérêts de l'utilisateur. Ceux-ci sont exprimés via des concepts hiérarchisés. La sélection des documents est réalisée via un ensemble de hiérarchies de concepts sur la collection qui représente la connaissance du domaine. L'originalité de l'approche réside dans l'utilisation du concept de cube de données présent dans les approches de type OLAP (On-Line Analytical Processing) avec des collections de documents. Chaque hiérarchie de concept est utilisée comme axe du cube et peut être explorée selon différents niveaux d'abstraction, reprenant ainsi les principes de forage des systèmes décisionnels. L'implantation de cette approche a été réalisée dans le système *DocCube* qui offre les fonctionnalités des hypercubes comme dans les systèmes OLAP.

Les techniques de visualisation on été aussi utilisées dans la découverte de connaissances, la plate-forme *Tétralogie* proposée dans [72] en donne l'exemple. Dans cette approche les auteurs offrent à l'utilisateur un outil interactif pour lui permettre non seulement la visualisation des données mais également sa participation au processus de découverte de connaissances. Ils ont proposé par exemple dans le cadre de la veille technologique et scientifique une méthode de détection et de suivi des équipes de recherche à travers les collaborations qui existent entre chercheurs.

Dans le domaine industriel des solutions décisionnelles, de nombreux outils de visualisation des données multidimensionnelles ont été développés. Citons par exemple *Explorer d'Oracle Express*[10], qui permet de manipuler directement les concepts de l'approche multidimensionnelle (fait, dimension, mesure ...) et de visualiser les données sous forme de tranches de cube de données. D'autres éditeurs de logiciels ont proposé des offres comme *DB2 OLAP Server*[11] de IBM, qui repose sur la technologie

[10] www.oracle.com/olap

[11] http://www-306.ibm.com/software/data/db2/db2olap/

d'*Hyperion Essbase* ou encore l'offre de *Microsoft OLAP Server*[12].

2.4.3 Manipulation des données multidimensionnelles

Les utilisateurs ont besoin d'outils de synthèse et de manipulation de données qui soient efficaces et conviviaux pour la prise de décision. Dans les cubes de données multidimensionnelles, les cellules fournissent des informations agrégées pour une mesure et selon plusieurs dimensions et les analystes emploient des opérateurs permettant de manipuler ces structures complexes.

Notons que dans la manipulation multidimensionnelle, les concepts intuitifs se sont propagés rapidement. Comme le montrent [19, 67], les termes utilisés par les opérateurs OLAP sont plus descriptifs que formels. Chaque analyste peut avoir plusieurs interprétations des résultats de ces opérateurs selon son point de vue et selon la synthèse qu'il veut avoir.

Plusieurs travaux de synthèse [2, 68, 99] ont classé les opérateurs de manipulation des données multidimensionnelles. Selon les systèmes et modèles proposés, nous pouvons réaliser une typologie des opérateurs de manipulation des données multidimensionnelles. Dans [55], l'auteur identifie deux classes d'opérations de manipulation de données multidimensionnelles. La première concerne les opérations *inter-cubes*, et la seconde concerne les opérations *intra-cubes* qui sont de deux types : celles qui ne changent que la présentation du cube et celles qui modifient le contenu du cube. L'auteur a proposé l'extension du modèle aux données floues ainsi qu'une algèbre de manipulation de ce modèle [56]. Dautres travaux ont étudié l'imprécision dans les données multidimensionnelles [78, 79].

D'autres travaux ont proposé une autre typologie différente des opérateurs de manipulation, selon l'intuition de l'utilisateur qui cherche à obtenir, soit une forte interaction de l'analyse en ligne de données en utilisant des opérateurs *de restructuration*, soit une hiérarchisation de l'information et donc il fait appel aux opérateurs *de granularité*, soit il peut simplement chercher à adapter les opérateurs relationnels en utilisant les opérateurs dit *classiques*. Les divers opérateurs définis pour la manipulation des données multidimensionnelles ne sont pas standardisés. Cependant nous nous baserons sur la dernière typologie présentée pour présenter ces différents opérateurs. La catégorisation des opérations OLAP en trois types : classiques, de granularité et de restructuration, est celle qui est retrouvée dans la majorité des travaux qui se sont intéréssés à la manipulation des cubes de données [2, 10, 43]. Les sections suivantes sont consacrées à la description des opérations de manipulation dans le cadre de l'analyse en ligne des données multidimensionnelles.

2.4.3.1 Les opérations classiques

Dans cette section nous présentons les opérations dites *classiques*, celles qui étendent les opérateurs de l'algèbre relationnelle. Les différents modèles de la littérature précisent que les opérations classiques opèrent sur le contenu des cubes et

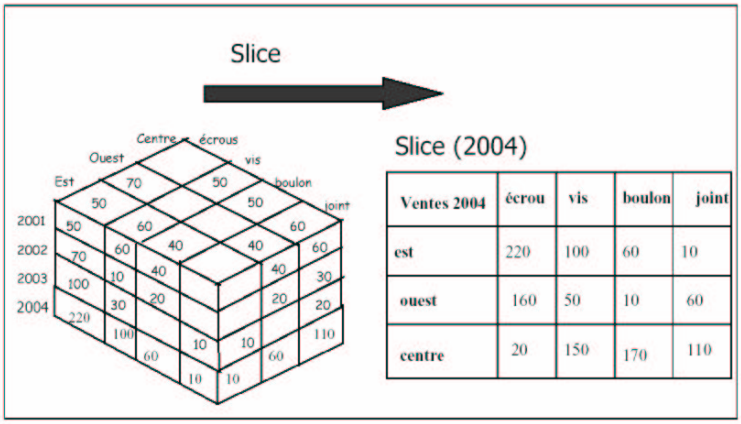
[12] http://www.microsoft.com/france/technet/produits/sql/7.0/a-olap-1005.mspx

laissent inchangée sa structure multidimensionnelle. Ainsi les opérations relationnelles classiques (sélection, projection, renommage, union, intersection, différence et produit cartésien) sur un cube sont directement traduites en opérations de l'algèbre relationnelle sur les relations sous-jacentes des cubes.

On peut retenir comme opération classique : la sélection, la projection, le produit cartésien, la jointure, les opérations ensemblistes (union, intersection et différence), la suppression d'une mesure, l'ajout d'une mesure calculée et le renommage.

Sélection. Si l'on considère la version relationnelle de l'opération de sélection, le principe est de retenir un sous-ensemble de données d'une relation qui contient des éléments satisfaisant les critères de cette sélection. Ce principe appliqué à un cube consiste à définir un sous-ensemble de ses données. On peut alors faire cette sélection de deux façons distinctes : la première, en faisant une sélection qui porte sur les mesures du cube (*slice*) et la seconde, en faisant une sélection sur ses membres (*dice*). La restriction ou le slice consiste à extraire de l'information résumée pour une certaine dimension. La figure 2.8 montre l'application d'un *slice* sur la dimension *année* avec la valeur 2004. On ne retient alors que les valeurs inférieures à un seuil.

– *Le slice :* consiste à couper une tranche du cube en faisant une sélection sur les cellules par des prédicats selon une dimension (voir figure 2.8).



Ventes 2004	écrou	vis	boulon	joint
est	220	100	60	10
ouest	160	50	10	60
centre	20	150	170	110

Figure 2.8 – Exemple d'un opérateur *Slice* tiré de [95]

– *Le dice :* consiste à extraire un sous-cube en faisant une sélection sur les dimensions. Cela revient à faire une restriction sur les dimensions et non plus par rapport à un critère sur la mesure. Comme illustré sur la figure 2.9, l'opération du dice est appliqué sur la dimension "année" avec (2001, 2002), la dimension "région" avec (est, ouest) et la dimension "pièces" avec (boulon, joint).

Projection. L'opération de projection consiste (en algèbre relationelle) à restreindre l'ensemble des attributs d'une relation. L'extension de cette opération au modèle multidimensionnel porte soit sur les dimensions, soit sur les niveaux de granularité.

– *La projection sur les dimensions.* Quand on projette un cube de n dimensions

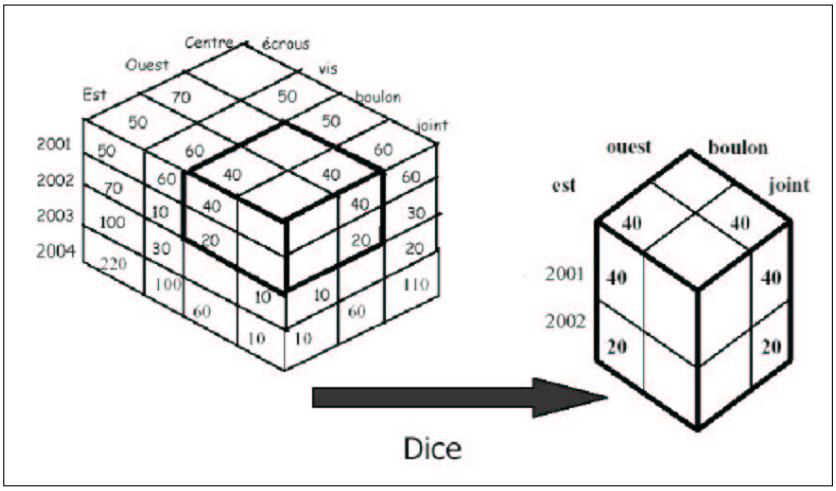


Figure 2.9 – Exemple d'un opérateur *Dice* tiré de [95]

sur $n - p$ dimensions, $1 \leq p \leq n$, le cube résultant de $n - p$ dimensions doit conserver l'unicité de l'association entre la référence et le contenu des cellules.

– *La projection sur les niveaux de granularité.* Cette projection permet sur un cube de données de choisir un seul niveau du cube de données à représenter. Dans ce cas le nombre de dimensions ne change pas. Ainsi sur la figure 2.10 le cube (illustré à gauche) peut être obtenu à partir de la tranche du cube (illustrée à droite), si on projette au plus haut niveau de la dimension *temps*.

On notera que la combinaison des opérations de sélection et de projection est souvent regroupée sous le terme *slice and dice* dans la littérature OLAP.

Jointure. La jointure consiste à combiner deux cubes par rapport à des dimensions. Elle permet comme dans le modèle relationnel d'associer des données issues de différentes relations disposant de certains champs ou des méta-données en commun dans le but d'aboutir à des relations plus riches que celles d'origine si on les considère chacune à part.

Fusion. La fusion consiste à fusionner les dimensions d'un cube selon une fonction d'agrégation.

Ajout, suppression et renommage. Ces opérations permettent à l'utilisateur d'ajouter ou de supprimer des mesures calculables à partir des données d'un cube OLAP ou encore de les renommer.

Opérations ensemblistes. Nous trouvons aussi dans la manipulation des cubes les opérateurs ensemblistes d'union, d'intersection et de différence.

2.4.3.2 Les opérations de granularité

Ces opérations agissent sur la granularité de l'observation des données. Elles guident la navigation entre les différents niveaux d'abstraction. Ces opérations nécessitent des informations non contenues dans le cube pour passer d'une représentation initiale à une représentation de granularité différente.

Généralisation

– *Forage vers le haut, pliage ou roll-up.* L'opération de *roll-up* consiste à représenter les données du cube à un niveau de granularité supérieur conformément à la hiérarchie définie sur une dimension. L'opérateur de Roll-up est basé sur la hiérarchie de dimension, il s'agit des différents niveaux de granularité de chaque dimension, par exemple celle qui peut décrire la dimension *produit* en plusieurs niveaux *écrous, vis, boulon et joint.* Cette opération revient à généraliser les valeurs d'un attribut à des concepts de niveau supérieur. On peut par exemple calculer la moyenne sur plusieurs valeurs de dimensions.

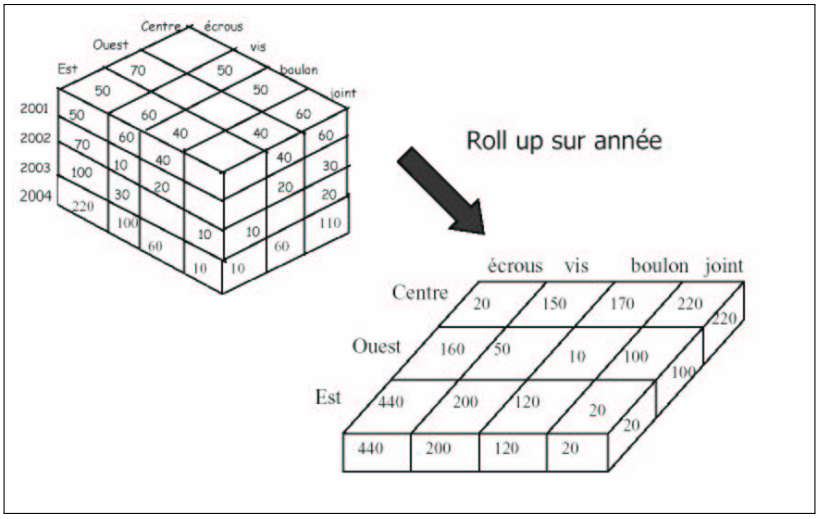


Figure 2.10 – Exemple d'une opération de Roll-up (tiré de [67])

– *DataCube.* Le *datacube* d'après [67] est l'opération qui décrit la consolidation d'un cube (i.e. calcul de tous les agrégats suivant tous les niveaux d'une dimension). Cette opération est considérée comme une généralisation de l'opération du roll-up.

Spécialisation

Le forage vers le bas, le dépliage ou le drill-down. L'opération de *drill-down* consiste à représenter les données du cube à un niveau de granularité inférieur, donc sous une forme plus détaillée.

L'application de ces opérations de forage vers le haut ou vers le bas nécessite de connaître des informations sur les valeurs des cellules du résultat car ces informations ne sont pas calculées à partir du cube général. Pour le *roll-up*, on doit connaître la fonction d'aggrégation utilisée, tandis que pour le *drill-down* on doit connaître la répartition des données à des niveaux de granularité plus fins.

2.4.3.3 Les opérations de restructuration

Dans [67], l'opération de restructuration est définie comme suit :

Definition 2.13 (Opération de restructuration)**.** *Une opération de restructuration est toute opération élémentaire bijective de changement de point de vue sur un cube. Ainsi, tout cube par restructuration d'un cube initial contient les informations nécessaires et suffisantes pour générer le cube initial par restructuration réciproque.*

Les opérations de restructuration sont alors considérées comme des opérations qui changent seulement la structure d'un cube sans toucher à son contenu. Les principales opérations de restructuration existant dans les systèmes OLAP, sont présentées ci-après.

Rotation. La *rotation* (ou l'opérateur rotate) est une opération de restructuration élémentaire qui consiste à faire effectuer à un cube une rotation ou un *pivot* autour d'un des trois axes passant par le centre de deux faces opposées, de manière à présenter un ensemble de données d'une face différente du cube. On note qu'un cube de dimension n a $n(n-1)$ vues. Pour afficher le cube sous un autre angle on le fait pivoter autour d'un de ses axes. Marcel [68] spécifie l'opération de rotation en opération typiquement tridimensionnelle qui n'a pas d'incidence sur le nombre de dimensions de la représentation adoptée pour le cube, et qui s'attache à rendre l'aspect tridimensionnel d'un cube dans un plan. Seules trois faces étant représentées, l'opération *rotate* s'apparente à une opération de sélection qui ne serait pas guidée par un choix de membres, mais par un choix de faces.

Permutation. L'opération de *permutation* ou *switch* consiste à inverser deux dimensions pour permuter deux tranches du cube. Cette opération consiste à interchanger les membres d'une dimension. Dans cette opération certaines informations se retrouvent alors placées l'une à côté de l'autre, ce qui facilite la découverte de corrélations entre attributs et peut aider à l'interprétation pour les analystes et les décideurs.

Division ou éclatement. L'opération de division (ou l'opérateur *split*) consiste à représenter chaque tranche du cube sous une forme tabulaire. Elle permet ainsi de passer d'une représentation tridimensionnelle d'un cube à sa représentation sous la forme d'un ensemble de tables. Cette opération permet de réduire le nombre de dimensions d'une représentation. Sa généralisation permet par exemple de découper un hypercube de dimension 4 en cubes. On peut alors réduire le nombre de dimensions et s'intéresser à un seul axe. Le nombre de tables résultant d'un *split* dépend du nombre de membres d'une dimension contenu dans le cube initial.

Emboîtement. L'opération d'emboîtement (ou l'opérateur *nest*) consiste à imbriquer les valeurs d'un paramètre de dimension avec un autre paramètre pour avoir une présentation bidimensionnelle. L'un des intérêts de cette opération est qu'elle permet de grouper sur une même représentation bidimensionnelle toutes les informations (mesures et membres) d'un cube ou hypercube, quelque soit le nombre de

ses dimensions. L'opération réciproque, *unnest*, reconstitue une dimension séparée à partir des membres imbriqués.

Enfoncement. Cette opération appelée aussi *push*, consiste à combiner les positions ou valeurs d'un paramètre d'une dimension aux mesures du fait et donc de transformer un paramètre en mesure. L'opération inverse de *retrait* ou *pull* permet de transformer une mesure en paramètre en changeant le statut de certaines mesures de l'hypercube pour constituer une nouvelle dimension. On constate que ces deux dernières opérations permettent de traiter de façon similaire mesures et membres.

Factualisation. Cette opération appelée aussi *fold* traduit en français dans [95] par la factualisation consiste à transformer une dimension en mesure(s). Elle permet de transformer en mesure l'ensemble des paramètres d'une dimension. L'opération inverse de *paramétrisation* (traduction de *unfold*) permet de transformer une mesure en paramètre dans une nouvelle dimension.

2.5 Conclusion

Les différentes opérations présentées précédemment sont considérées comme les opérations élémentaires pour l'analyse en ligne des données. La majorité des produits implémentant les fonctionnalités OLAP couvrent ces opérations. Ces produits doivent offrir à l'utilisateur des outils de manipulation intuitifs comme le souligne E. F. Codd dans les douze règles présentées en section 2.2.3 de ce chapitre.

D'un autre côté, [16] met l'accent sur la nécessité de définir plusieurs langages de manipulation de données à différents niveaux d'abstraction. Ces langages doivent offrir à l'utilisateur, qui désire effectuer des traitements complexes, la possibilité d'avoir un langage formel de haut niveau. Dans le travail de synthèse de P. Marcel [67], l'auteur fait une étude approfondie sur les langages formels de manipulation des données multidimensionnelles proposés dans la littérature [2, 16, 34, 63, 78].

Le chapitre suivant vise à dresser un état de l'art des méthodes et approches qui s'intéressent à résumer les données volumineuses dans le même principe que celui des systèmes d'information décisionnels. Ceci dans le but de présenter aux décideurs et aux analystes, des données sous une forme réduite mais riche en connaissances.

Chapitre 3

La compression sémantique des données

Au lieu de répondre très bien, je réponds très, et le bien qui me reste, je vais le porter à ma banque.

— Raymond Devos, 1992.

Avec la mondialisation, Internet et la part grandissante de l'informatique, les centres de calcul traitent une quantité de données de plus en plus importante. Cette problématique de masse de données s'étend à toutes les disciplines scientifiques ainsi qu'aux données issues des activités commerciales et industrielles ou encore des études et observations dans le domaine des sciences humaines et sociales. Cette masse de données provient par exemple du nombre élevé des dispositifs expérimentaux (par exemple en biologie) ou encore de la grande quantité des informations collectées sur des individus (les consommateurs par exemple) ou sur des produits industriels.

C'est pour ces raisons que les aspects relatifs aux grandes masses de données constituent actuellement des axes de recherche de grande importance en ce qui concerne le stockage, le traitement ou la présentation de ces données, ainsi que la meilleure façon de réduire ces grands volumes de données.

Nous présentons dans ce chapitre un ensemble d'approches qui tendent à vouloir produire des résumés à partir de données structurées. Cette présentation est organisée selon les différentes thématiques liées à la compression sémantique des données.

3.1 La compression des données

Traditionnellement, la compression de données concerne l'optimisation de l'utilisation des ressources systèmes comme l'espace de stockage. Nous considérons que cette compression de données est « syntaxique » dans le sens où il s'agit de compresser un fichier en réduisant le nombre de digits binaires nécessaires pour l'enregistrer. Une variété de méthodes de compression avec ou sans perte ont été développées pour certaines formes de données (image, audio, vidéo ...). Un grand nombre des méthodes de compression sans perte ou réversibles se basent sur des techniques statistiques ou des dictionnaires. Parmi les méthodes basées sur des techniques de codage statistique, la méthode de compression proposée par David Albert Huffman en 1952 [41].

La méthode de compression Huffman consiste à diminuer au maximum le nombre de bits utilisés pour coder un fragment d'information. Prenons l'exemple d'un fichier de texte : le fragment d'information sera un caractère ou une suite de caractères. Plus le fragment sera grand, plus les possibilités seront grandes et donc la mise en œuvre complexe à exécuter. L'algorithme de Huffman se base sur la fréquence d'apparition d'un fragment pour le coder : plus un fragment est fréquent, moins on utilisera de bits pour le coder. Les autres méthodes de compression sans perte, dites à dictionnaire, sont basées sur l'algoritme *LZW* publié dans un article de Jacob Zif et Abraham Lempel en 1977 [109] dont le principe est d'utiliser un dictionnaire dynamique qui contient des motifs du fichier traité.

D'un autre côté, il existe des méthodes de compression de données dites « sémantique », qui sont motivées par l'accès, l'analyse et l'exploration d'un grand volume de données qui ne cesse d'augmenter. En effet, en raison de la nature exploratoire de beaucoup d'applications d'analyse de données, il existe plusieurs scénarios dans lesquels il est difficile d'obtenir une réponse exacte à une requête complexe, et les analystes peuvent préférer une réponse rapide et approximative, tant que le système peut garantir un certain seuil de tolérance d'erreur. Cette réponse rapide peut être obtenue en accédant à des espaces de stockage qui contiennent des vues résumées de la base de données initiale. Dans ce chapitre, nous écartons le principe de la compression syntaxique. Notre intérêt est de présenter les méthodes qui fournissent une vue synthétique d'une grande masse de données.

3.1.1 Travaux sur la compression sémantique des données

Les travaux sur la compression sémantique de données sont consacrés à l'étude de la construction de vues qui regroupent le résumé d'un ensemble de données à des niveaux d'abstraction élevés, ceci en conservant la sémantique des données.

La figure 3.1 montre une catégorisation des approches de compression sémantique qui seront présentées dans les sections qui suivent. Ces approches ont toutes comme objectif commun la réduction de grandes masses de données structurées en respectant la sémantique de ces données. Sans prétendre à l'exhaustivité, ce recensement des différentes méthodes utilisées pour la compression sémantique, rassemble les principales approches de la littérature du domaine.

Nous proposons un classement selon deux grandes familles : les méthodes statistiques, et les méthodes basées sur les modèles. Pour la première catégorie on peut diviser les méthodes statistiques en celles qui visent à réduire le nombre d'instances, celles qui réduisent le nombre d'attributs et celles qui opèrent des calculs d'agrégats (comme les systèmes OLAP). La deuxième famille concerne les méthodes basées sur les modèles. Elle prend en considération les méthodes issues du domaine de la fouille de données de même que les méthodes basées sur les méta-données comme les méthodes d'induction par attribut. Les sections suivantes sont dédiées à la présentation de ces différentes approches.

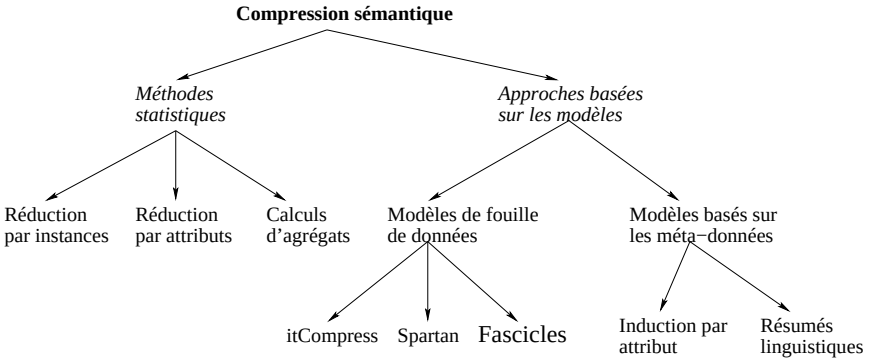


Figure 3.1 – Typologie des approches de compression sémantique de données

3.2 Méthodes de compression sémantique

3.2.1 Les méthodes statistiques

Les méthodes statistiques font partie des processus d'analyse de données qui visent à générer des résumés à partir d'une grande masse de données. Ces approches s'appuient sur l'extraction des informations qui sont jugées intéressantes parce qu'elles sont souvent représentées par un grand nombre de données. Nous présentons ici les trois types de méthodes statistiques utilisées dans l'analyse de données :

- Les méthodes verticales basées sur la réduction du nombre d'instances.
- Les méthodes horizontales basées sur la réduction du nombre d'attributs.
- Les méthodes du calculs d'agrégats.

3.2.1.1 Résumé du nombre d'instances

Le résumé des instances d'une base de données, du point de vue statistique, décrit les aspects quantitatifs des données comme le nombre de n-uplets, la distribution des valeurs ou encore les corrélations existantes entre ces valeurs. Cette approche utilise des techniques statistiques en calculant des indicateurs tel que la moyenne, l'écart type, la variance ou la médiane. Le but est de pouvoir caractériser la distribution des observations autour d'un indicateur comme la moyenne par exemple. Parfois la moyenne n'est pas suffisante pour calculer la dispersion des valeurs, dans ce cas le calcul de la variance peut répondre à cette exigence. Les différentes observations réalisées à partir d'un ensemble de données, permettent de construire un modèle à partir d'un ensemble restreint de ces observations. Le modèle construit devrait affirmer avec suffisamment de confiance que son application aura un effet déterminé sur n'importe quel individu de la population. Un tel modèle peut être réalisé à partir d'un échantillon, une technique statistique très utilisée dans l'analyse des données.

Echantillonnage

Généralement, pour étudier les caractéristiques d'une population on peut choisir entre deux approches. La première est d'étudier chaque unité de la population. Ce processus est appelé en statistique *l'énumération complète*. La deuxième est d'étudier les caractéristiques de la population en examinant une partie de cette population, cette méthode est appelée *échantillonnage*. Théoriquement la première approche est plus efficace mais dans le cas des masses de données, son application devient très difficile.

Definition 3.1 (Echantillon)**.** *Un échantillon est un sous-ensemble d'une population sur lequel on effectue une étude statistique. Une étude sur un échantillon vise généralement à tirer des conclusions relatives à l'ensemble de la population.*

L'échantillonnage est une technique statistique qui sélectionne une partie d'une grande population afin de pouvoir étudier cette population et mieux connaître ses propriétés. Cette technique peut être utilisée pour résoudre des problèmes causés parfois par la haute dimensionnalité de l'espace de représentation des données. Elle est très utilisée dans le domaine de la fouille de données quand il s'agit de très larges volumes d'informations. La principale difficulté de l'échantillonnage réside dans la façon de créer un échantillon à partir d'un grand volume de données, qui soit représentatif et qui puisse s'y substituer lors de futurs traitements. Il existe plusieurs méthodes d'échantillonnage comme la méthode aléatoire simple, aléatoire stratifiée ou en grappes. Ces méthodes peuvent être de type probabiliste ou non probabiliste. Un échantillonnage est dit probabiliste quand sa procédure de sélection se base sur la théorie des probabilités. On cite pour ce cas l'échantillonnage aléatoire simple ou stratifié. En revanche, quand l'échantillonnage est fait sans utiliser la théorie des probabilités on dit qu'il est non probabiliste. Ceci signifie que les éléments de l'échantillon ne dépendent d'aucun résultat de probalilité dans leur procédure de sélection. Le choix de la méthode à utiliser est une phase très importante dans un processus d'échantillonnage. Ce choix dépend de plusieurs éléments comme la définition de la population à étudier ou la détermination de la taille de l'échantillon. Nous donnons ici l'exemple d'une méthode d'échantillonnage appelée *méthode des quotas* pour montrer comment un échantillon peut être construit. Nous choisissons cette méthode parce qu'elle définit les échantillons sous forme de classes que nous pouvons considérer comme une vue réduite d'un ensemble de données.

Echantillonnage par quotas. Cette méthode consiste à classer préalablement les individus en n classes $C_1, \ldots, C_n$. La représentativité de chaque classe par rapport à la population complète est liée alors au cardinal de chaque classe. La difficulté est dans le choix des classes car, constituées selon des critères non indépendants de variable aléatoire [1], elles peuvent générer une sur (sous)-représentativité de certaines des valeurs de cette dernière. Cette méthode est employée principalement sur une

[1] une variable aléatoire est une fonction définie sur l'ensemble des résultats possibles d'une expérience aléatoire, telle qu'il soit possible de déterminer la probabilité pour qu'elle prenne une valeur donnée ou qu'elle prenne une valeur dans un intervalle donné.

collection de classes connues a priori, mais il est parfaitement possible de l'envisager sur une collection issue d'une classification automatique.

L'objectif ultime de l'échantillonnage est de faire des inférences à propos des intérêts de la population. La performance de n'importe quelle estimation faite à partir d'un échantillon dépend de la méthode choisie pour faire l'échantillon et de celle qui permet de calculer l'estimation à partir de la base de données de l'échantillon. L'évaluation d'un estimateur se fait selon l'absence de biais et la faible variance de l'échantillon. Il est à noter qu'un échantillon ne donne pas exactement les mêmes estimations correctes que sur la vraie population, la marge de risque reste à limiter.

3.2.1.2 Résumés du nombre d'attributs

La seconde approche pour résumer des données en utilisant des méthodes statistiques est le résumé du nombre d'attributs. Cette approche consiste à réduire la complexité des données en ne gardant qu'un nombre restreint d'attributs, les plus importants par exemple. Dans cette approche nous présentons les différentes méthodes d'analyses factorielles. Les analyses factorielles trouvent tout leur intérêt pour la compréhension des tableaux de grande dimension, plusieurs dizaines ou centaines de lignes et de colonnes, que les traitements statistiques classiques ne peuvent interpréter de façon globale. Ces méthodes ont été également utilisées pour la réduction du nombre d'attributs d'une base de données.

Méthodes d'analyse factorielles

Le principe de ces méthodes est de proposer une synthèse de la représentation des données en réduisant le nombre d'attributs afin de faciliter l'interprétation des relations ou corrélations existantes entre les données. Ainsi ces méthodes visent à décrire l'information globale fournie par plusieurs variables décrivant un ensemble d'individus. Elles permettent de distinguer les individus entre eux en tentant de trouver les facteurs qui expliquent cette distinction. Un des intérêts majeurs de ces analyses factorielles est de fournir une méthode de représentation d'une population décrite par un ensemble de caractères dont les modalités sont quantitatives (mesures continues), pour une ACP, ou qualitatives (pour une AFC). Ainsi, nous présentons ici les plus courantes des méthodes factorielles : ACP, AFC et l'analyse discriminante.

Analyse en Composantes Principales (ACP). Cette méthode a comme objectif de chercher les meilleurs axes de projection par combinaison linéaire d'attributs quantitatifs. Ces axes devront faciliter l'analyse des données puisqu'ils doivent montrer la meilleure représentation graphique des individus sur un plan bidimensionnel. Elle est utilisée pour sélectionner un nombre restreint de variables synthétiques qui sont les composantes principales. L'intérêt de cette méthode est qu'elle est automatique et permet de détecter les hypothèses de dépendance entre les attributs, même si les axes produits restent difficile à interpréter pour l'utilisateur.

Analyse Factorielle des Correspondances (AFC). Cette méthode s'applique initialement aux tableaux de contingence qu'on peut obtenir en croisant deux variables

qualitatives. La validité de cette méthode s'étend à tous les tableaux qui satisfont deux conditions. La première impose que les données du tableau sont toutes positives, et la seconde que toutes les grandeurs du tableau soient de même nature. L'objectif de réaliser une AFC est de rechercher les corrélations entre deux attributs nominaux, cela revient à appliquer une ACP avec une symétrie totale sur deux nuages de points projetés.

L'analyse discriminante. L'objectif de cette méthode est de mettre en évidence une relation entre un attribut nominal et plusieurs attributs numériques, dans le cadre de l'explication d'une répartition en classes connues. Une classe contient un groupe d'individus qui sont décrits sur un ensemble d'attributs et sont identifiés comme appartenant à cette classe particulière. Cette méthode permet de déterminer les meilleurs axes pour l'explication de ces classes en recherchant les combinaisons linéaires des axes de départ qui fournissent la meilleure séparation entre deux classes.

Les méthodes factorielles tiennent une place primordiale dans les méthodes d'analyse de données. Elles ont largement démontré leur efficacité dans l'étude des grandes masses d'information. Leur force vient du fait qu'elle permettent la confrontation de nombreuses informations ce qui est plus riche en renseignements qu'un examen séparé. Ainsi elles peuvent être utilisées pour la description d'une population afin de mieux l'analyser. Les méthodes ACP et AFC peuvent être utilisées dans le cas où des données sont décrites par un ensemble de variables qui ont toutes la même importance et jouent le même rôle. L'analyse discriminante peut être utilisée quand il s'agit d'expliquer des phénomènes au sein des données en fonction d'autres, ce qui permet de prévoir parfois des résultats autrement imprévisibles.

3.2.1.3 Calcul d'agrégats

Les approches de compression sémantique s'appuyant sur le calcul d'agrégats se retrouvent dans les BDS (Bases de Données Statistiques) et dans les systèmes OLAP.

Les BDS apparues au début des années 1970, ont comme principales caractéristiques de collecter des informations et de fournir aux utilisateurs des outils pour l'analyse des données. L'analyse provient de traitements statistiques simples ou complexes. Dès les années 80, les travaux sur les BDS [58] sont consacrés à l'étude de la construction, de la maintenance et de la pertinence d'agrégats calculés à partir de fonctions statistiques. Ceci est motivé par le besoin identifié par les statisticiens, de gérer et analyser efficacement les grandes masses de données se trouvant dans des bases de données. Ils choisissent de représenter les données et de les traiter d'une manière résumée, par groupe d'individus ou par sous-ensembles. Les données dans les BDS sont considérées selon deux types de valeurs : celles qui représentent les paramètres afin d'identifier des catégories de données, et celles qui représentent les variables qui sont appelées également des mesures. Les opérations d'agrégation et de désagrégation sont définies pour représenter les informations à différents niveaux de granularité. Ces agrégations réduisent aussi l'espace de stockage des données, comme le fait de ne stocker que des codes à la place des valeurs des catégories. Le

calcul d'agrégats dans les BDS a permis la création des tables résumées décrivant la distribution multivariée des données [29].

OLAP

Le contexte OLAP, détaillé dans le chapitre précédent, a repris de nombreux concepts des bases de données statistiques. Les relations du contexte OLAP avec les bases de données statistiques ont été étudiées, notamment par [94], mettant en évidence les similarités de ces deux modèles. Les concepts communs à ces deux approches sont la multidimensionnalité, les hiérarchies et les attributs de résumés (appelés mesures dans le modèle OLAP). Dans les deux contextes, on trouve des données de base (également appelé niveau micro), des données de niveau agrégé (également appelé niveau macro), et des méta-données (hiérarchies par exemple). Les données sont distinguées selon qu'elles sont des mesures ou des variables. Si les bases statistiques ne présentent souvent à l'utilisateur que les données de niveau macro, les outils OLAP tendent à travailler à partir des données de niveau micro.

Quotient cube

Des travaux [52, 53, 54] ont été menés pour proposer des constructions de résumés de datacubes qui soient plus facilement visualisables tout en gardant les propriétés sémantiques du treillis du cube [2] permettant justement de voir les tendances cachées dans le cube de données. Une approche proposée consiste à regrouper en classes les sommets du treillis en fonction des valeurs qui leur sont associées : deux sommets font partie d'une classe s'ils ont la même valeur. Cette approche est appelée *Quotient Cube*, elle a été introduite par V. S. Lakshmanan et al. dans [52] et implémentée dans le système SOCQUET [53, 54]. L'objectif de l'approche est de permettre à l'utilisateur une navigation similaire à celle qui serait possible au sein d'un cube classique, mais en se déplaçant à un niveau d'abstraction supérieur qui regroupe les cellules en régions homogènes.

Les méthodes basées sur des approches statistiques précédemment présentées ont une grande capacité à fournir des résumés de données à un utilisateur en lui proposant des outils de représentation graphique et d'analyse. Toutefois, la faiblesse de ces méthodes se situe dans leur incapacité à offrir à un décideur la possibilité de découvrir des connaissances exceptionnelles souvent ne concernant pas la majorité des individus. En effet, ces méthodes synthétisent de manière très concise un ensemble très important de données, en négligeant les connaissances relatives à des sous-ensembles particuliers des données à résumer. Certes les systèmes OLAP sont dotés d'outils de fouille de données mais ils restent toujours limités par l'effet de seuil et l'orientation sujet qui est définie en amont d'une étape de découverte de connaissance. Nous allons présenter dans la section suivante les méthodes de compression sémantique de données basées sur les modèles de fouilles de données ou sur les méta-données.

[2] Le quotient cube considère un cube de données par sa présentation sous forme d'un treillis de Galois.

3.2.2 Les méthodes basées sur les modèles

3.2.2.1 Modèles de fouille de données

Dans les méthodes basées sur les modèles de fouilles de données, nous étudions ici les différentes approches de classification utilisées pour la compression sémantique.

La classification des données sert à mettre en évidence des relations entre objets, ou individus, et entre objets et paramètres d'objets. Le processus de classification permet de construire une partition de l'ensemble des objets en classes relativement homogènes. Telle qu'elle est définie, la classification constitue un outil efficace pour la compression sémantique de données. Il nous paraît essentiel, pour réduire les données et construire des résumés basés sur des algorithmes de classification, de présenter les principales méthodes de classification supervisée et non supervisée, qui sont utilisés dans le cadre de la compression sémantique.

Classification supervisée

La classification supervisée considérée comme une approche d'apprentissage automatique peut être réalisée de deux façons. L'une dite empirique, considère un étiquetage préalable, résultat d'une expertise sur le domaine étudié, en exemples et contre-exemples des phénomènes à caractériser. L'autre, appelée constructive, procède par auto-apprentissage en capitalisant l'expérience des observations passées et supervisées en s'appuyant sur des connaissances de domaine pour initialiser le processus. Il existe de nombreuses méthodes d'apprentissage supervisé :

- k plus proches voisins,
- arbres de décisions,
- méthodes bayésiennes,
- réseaux de neurones,
- programmation génétique.
- ...

Deux méthodes d'apprentissage supervisé on été utilisées dans des systèmes de compression sémantique appelés *Spartan* et *Fascicles*. Ces deux systèmes s'appuient respectivement sur les réseaux bayésiens et la classification par la méthode des centres mobiles.

Méthode des réseaux bayésiens. Les réseaux bayésiens sont nommés ainsi d'après le théorème de *Bayes*. Cette méthode suppose l'indépendance des variables. L'idée est d'utiliser des conditions de probabilité observées dans les données. On calcule la probabilité de chaque classe parmi les exemples. Un classifieur basé sur le théorème de *Bayes* consiste à construire un tableau qui détermine la probabilité d'appartenance à une classe d'une donnée sur la base de combinaison des probabilités sur chaque attribut.

Theoreme 1 (Théorème de Bayes)**.** *Soit deux événements A et B, la probabilité a*

posteriori de l'événement A sachant la réalisation de l'événement B est :

$$P(A/B) = \frac{P(A \cap B)}{P(B)}$$

de même la probabilité à posteriori de l'événement B sachant la réalisation de l'événement A est :

$$P(B/A) = \frac{P(B \cap A)}{P(A)}$$

Les réseaux bayésiens modélisent un classifieur sous la forme d'un graphe paramétré qui représente la distribution conjointe d'un ensemble de variables. L'apprentissage automatique des réseaux bayésiens dans les bases de données vise à déterminer la structure du graphe dont les nœuds sont certains attributs qui sont mis en relation avec d'autres attributs par les arcs du graphe à l'aide de la détermination des probabilités, améliorant ainsi la précision du classifieur. Les réseaux bayésiens sont actuellement utilisés dans des domaines assez variés comme par exemple le diagnostic de panne, la modélisation d'utilisateur, l'aide à la décision médicale, ou bien encore la robotique [57, 59, 50]. Des méthodes ont été proposées pour la recherche du meilleur réseau [27]. Un réseau bayésien est défini comme suit :

Definition 3.2 (Réseau bayésien)**.** *Une structure $B = (\mathcal{G}, \theta)$ est un réseau bayésien si $G = (X, E)$ est un graphe acyclique dirigé dont les sommets représentent un ensemble de variables aléatoires $X = \{X_1, X_2, \ldots.., X_n\}$ et si $\theta_i = \left[P(X_i|X_{Pa(Xi)})\right]$ est la matrice des probabilités conditionnelles du nœud i connaissant l'état de ses parents $P_a(Xi)$ dans G.*

SPARTAN. Les réseaux bayésiens ont été utilisés dans le cadre de la compression sémantique de tables dans le système SPARTAN [6]. Dans ce système, un ensemble d'attributs dits *prédictifs* est utilisé conjointement avec un modèle reposant sur un réseau bayésien pour *prédire* les autres attributs. Il s'agit donc d'une compression horizontale puisque l'objectif est de réduire le nombre d'attributs mémorisés. La méthode de compression de SPARTAN exploite la sémantique des attributs afin de réduire le volume des tables. L'avantage de ce système consiste en son exploitation des corrélations prédictives entre les attributs d'une table et d'une tolérance à l'erreur qui est spécifiée par l'utilisateur. SPARTAN identifie des dépendances dans les données en construisant un réseau bayésien pour les attributs en question. Ces derniers participent à la construction d'un arbre appelé *Cart : Classification and Regression Tree*, qui résume la totalité des colonnes de la table.

Méthode des centres mobiles. Les algorithmes de classification par les centres mobiles, permettent de créer des classes regroupées autour d'un noyau. Un ensemble de données Ω sera classé en k classes : $C_1, \ldots C_k$. L'algorithme le plus connu dans cette catégorie est le *k-Means* [25, 65] et sa version améliorée appelée méthode des nuées dynamiques [24]. Cette méthode cherche à mettre à jour une partition en k classes (k donné) par divisions successives de l'ensemble de données initial. Chaque

classe est représentée par un noyau qui peut être un point (individu), un ensemble de points ou un espace. La partition optimale n'est atteinte qu'exceptionnellement du fait du choix arbitraire des k noyaux initiaux.

ItCompress : Dans le cadre de la compression sémantique de base de données, une variante de la méthode des centres mobiles a été utilisée par l'algorithme *ItCompress* dans [45]. La figure 3.2 donne une présentation d'une table selon l'exemple extrait de [45].

age	salaire	patrimoine	confiance	genre
20	30 000	25 000	faible	masculin
25	76 000	75 000	bon	féminin
30	90 000	200 000	bon	féminin
40	100 000	175 000	faible	masculin
50	110 000	250 000	bon	féminin
60	50 000	150 000	bon	masculin
70	35 000	125 000	faible	féminin
75	15 000	100 000	faible	masculin

(a) Table T d'origine

ERId	Bitmap	Valeurs exceptionnelles
2	01011	20, 25 000
1	11011	75 000
1	11111	
1	01100	40, faible, masculin
1	01111	50
1	01110	60, masculin
2	11110	féminin
2	11111	

(b) Table compressée T_c

ERId	age	salaire	patrimoine	confiance	genre
1	30	90 000	200 000	bon	féminin
2	70	35 000	125 000	faible	féminin

(c) Enregistrements représentatifs

Figure 3.2 – Compression de tables par *ItCompress*

`T` est la table d'origine (voir figure 3.2(a)), elle est représentée par la table $\mathtt{T}_c$ (figure 3.2(b)) au moyen de:

1. *ERId :* C'est l'identifiant d'un enregistrement de référence, il représente un enregistrement existant considéré comme typique de l'enregistrement à décrire.
2. *Valeurs booléennes :* C'est une liste de valeurs qui permet de savoir pour chaque attribut si la valeur de l'enregistrement de référence est satisfaite ou non. Une valeur dite proche est acceptable grâce à un niveau de tolérance qui doit être fixé.

3. *Valeurs exceptionnelles :* C'est une liste de valeurs qui s'avèrent insatisfaisantes dans la mesure où la valeur réelle de l'enregistrement courant doit être représentée correctement par l'enregistrement de référence sur des attributs.

Dans cet algorithme l'utilisation de la méthode des centres mobiles se fait pour rechercher les k enregistrements représentatifs. Ces enregistrements sont considérés comme le noyau, et à l'aide des enregistrements les plus proches de ces noyaux on peut construire les k classes. Le nombre d'attributs dont la valeur n'est pas correctement représentée par l'enregistrement de référence est utilisé dans l'algorithme de *ItCompress* comme une mesure de distance.

Classification non supervisée

Dans cette seconde approche de classification dite non supervisée, les classes possibles ne sont pas connues à l'avance, et les exemples disponibles sont non étiquetés. Le but est donc de regrouper (dans un même groupe) les objets considérés comme similaires, pour constituer les classes (créer, par exemple, différents groupes de textes, à partir d'un ensemble de textes de tous genres, selon la similarité de leur contenu). Dans le domaine de l'extraction de données et spécialement dans l'analyse d'ensembles de données, une des techniques utilisée est la recherche des règles d'association qui visent a prédire la classe étant donné un ensemble de conditions. Les conditions sont des paires de valeurs d'attributs, appelés aussi *itemsets*.

Règles d'association. L'utilisation des règles d'association est une méthode de classification non-supervisée qui consiste à déterminer les valeurs associées parmi les données. L'exemple le plus courant de l'utilisation de cette méthode est celui dit du *panier de la ménagère*, où le but est de déterminer les articles dont les achats sont simultanés. Par exemple si un client achète du poisson et du citron il achète aussi du vin blanc. Cette technique est utilisée dans plusieurs domaines comme l'analyse du comportement des consommateurs, les services bancaires ou les services de télécommunication.

$$\textit{Une règle d'association est de type : } X \rightarrow Y \textit{ signifie : si } X \textit{ alors } Y,$$

où la prémisse X et la conclusion Y sont des conjonctions de propriétés définies comme des attributs booléens. La propriété de décomposition des règles d'association permet de considérer que la conclusion Y n'est formée que d'un seul attribut. Pour valider une règle d'association on utilise le *support* ou la *confiance*.

Definition 3.3 (Support d'une règle d'association)**.** *Le support* ***supp*** *d'une règle d'association $r : X \rightarrow Y$ est la fréquence d'apparition simultanée des attributs X et Y. On le note :*

$$supp(r) = \frac{|XY|}{|R|}$$

où $|XY|$ est le nombre des n-uplets qui vérifient à la fois X et Y, et $|R|$ est le cardinal de la relation R sur laquelle est évaluée la règle r.

Definition 3.4 (Confiance d'une règle d'association)**.** *La confiance* ***conf*** *d'une règle d'association $r : X \rightarrow Y$ est le rapport entre $|XY|$ (le nombre des n-uplets qui vérifient à la fois X et Y) et $|X|$ (le nombre des n-uplets qui vérifient seulement X). La confiance est notée :*

$$conf(r) = \frac{|XY|}{|X|}$$

La découverte des règles d'association dans les données revient à déterminer les ensembles fréquents dans des matrices de co-occurrence. Le problème est combinatoire lorsqu'il s'agit de tester les multiples conjonctions d'attributs booléens, et de nombreuses optimisations sont envisagées. La méthode de *l'élagage par support minimum* est la plus courante, elle ne considère que les règles dont le support a atteint un seuil fixé. L'objectif principal pour l'extraction de règles d'association est de donner une information synthétique sur une partie de la base de données; ce qui n'empêche pas l'existence de règles dites triviales ou inutiles. Les premières ne donnent aucune information et les secondes sont trés difficiles à interpréter. Agrawal et al. [5] ont proposé *APRIORI*, un algorithme considéré comme une référence dans ce domaine. Il a été conçu dans un contexte d'extraction de connaissances pour des applications commerciales spécialement pour identifier les règles d'association du problème du *panier de la ménagère*.

Fascicles. Les règles d'association ont aussi été utilisées dans le cadre de la compression sémantique d'une table. Ce sont H. V. Jagadish et al. qui proposent dans [44] l'utilisation d'une forme étendue des règles d'association pour la compression d'une table. Ils utilisent pour cela la notion de *fascicles*, qui correspond à un sous-ensemble des enregistrements de la table dont une partie au moins des attributs ont des valeurs voisines au sens d'une certaine tolérance définie pour chaque attribut. La compression avec perte de la table est alors réalisée en remplaçant un fascicle par un unique enregistrement qui réalise le compromis des valeurs observées chez les enregistrements qui composent ce fascicle. Le problème rejoint, par de nombreux aspects, celui traité par les algorithmes de R. Agrawal et al. [3, 5] pour la recherche d'ensembles fréquents. L'extension se situe au niveau de la souplesse autorisée dans le nombre d'attributs voisins, ce qui accroît très fortement l'espace de recherche.

La classification floue

La théorie des sous-ensemble flous[3] offre une meilleure explication et une précision plus grande, pour représenter l'appartenance des objets aux classes. En effet la classification floue entre dans le cadre des méthodes de partitionnement [49], c'est-à-dire qu'on doit préalablement fixer le nombre de classes. Cependant, contrairement à la classification traditionnelle dite *nette* ou *dure*, la classification floue permet d'estimer des degrés d'appartenance de chaque individu à une classe. Par conséquent, un individu bien représenté aura un degré d'appartenance proche de l'unité à une classe et proche de zéro pour les autres classes, alors qu'un individu mal représenté aura

[3] une annexe est consacrée à rappeler les éléments de base de cette théorie.

plutôt des degrés d'appartenance uniformément faibles sur toutes les classes. Ce type de méthodes permet d'affiner l'interprétation des résultats quant à la contribution de chaque individu pour la construction d'une classe. Cette représentation d'appartenance des objets aux classes est plus conforme à ce qu'on attend d'un processus robuste et intelligent de classification. E. H. Ruspini [89] est le premier à avoir introduit le concept de sous-ensembles flous en classification suivi d'un bon nombre de chercheurs proposant d'appliquer à la classification différentes méthodes issues de la logique floue. Deux grandes familles de méthodes peuvent être distinguées, selon la manière dont la logique floue est utilisée. La première correspond aux méthodes issues directement de la logique classique. Ce sont en fait des versions *fuzzifiées* des méthodes classiques, utilisant des sous-ensembles flous pour modéliser des classes d'objets. On peut citer l'extension au flou de la méthode des centres mobiles proposée par James.C. Bezdek dans [8] ainsi que l'algorithme des *fuzzy c-means* [4]. La deuxième famille des méthodes repose sur les relations floues entre les valeurs d'un vecteur d'attributs et la classe à laquelle appartient un prototype. Elles permettent de représenter l'appartenance partielle des objets à plusieurs classes, ce qui est conforme a ce que peut fournir un processus d'apprentissage automatique en résultat.

Nous avons abordé dans cette section les méthodes basées sur les modèles pour la compression sémantique. Nous avons présenté ceux qui s'appuient sur des modèles de fouilles de données dont la classification supervisée, non supervisée et la classification floue. Dans la section suivante nous nous intéressons aux modèles basés sur les méta-données.

3.2.2.2 Modèles basés sur les méta-données

Les modèles basés sur les méta-données ont comme objectif la caractérisation de données en fournissant une généralisation concise et succincte d'un ensemble de données. Ces méthodes sont basées sur la description des concepts et sont considérées comme des approches de fouille de données descriptive dans le sens où elles décrivent les concepts ou les données pertinentes pour l'analyse sous une forme concise et générale. Dans cette section nous présentons deux modèles, de réduction de grands volumes de données, basés sur les méta-données : l'induction par attribut et les résumés linguistiques.

Induction par attribut

La méthode d'induction par attribut a été proposée par J. Han et al. dans [36], et a été développée pour la découverte de connaissance dans les bases de données. Le principe de cette méthode est d'abord la généralisation et puis l'agrégation par fusion des tuples généralisés identiques en conservant leurs effectifs. La généralisation s'appuie sur une connaissance de domaine ordonnée, dont les termes supérieurs représentent des notions générales regroupant un ensemble de termes inférieurs. Cette connaissance de domaine peut être une hiérarchie de concepts, préalablement disponible sur chaque attribut. L'objectif de cette hiérarchie est de décrire le passage des

concepts les plus spécifiques, correspondant aux valeurs d'attribut dans la base de données, aux concepts les plus généraux. Cette hiérarchie permet la réécriture des données d'une relation en utilisant le premier niveau de généralisation. Ce processus peut être répété jusqu'à l'obtention d'une nouvelle relation dont la taille ne dépasse pas un seuil qui a été fixé en amont. Différents types de connaissances peuvent être découverts efficacement en utilisant l'induction par attribut, y compris des règles caractéristiques, des règles discriminantes ou des règles quantitatives. Le processus d'induction par attribut a fait par la suite l'objet d'une implémentation appelée DBLEARN [37] et puis DBMINER [38] que nous présentons ci-après.

DBLEARN : dans [37], J. Han et Y. Fu proposent l'utilisation de DBLEARN pour l'amélioration des connaissances de domaines par l'adaptation dynamique du processus dans le choix des niveaux de généralisation utilisés. L'idée générale est d'éviter de sur-généraliser les valeurs des attributs qui couvrent de nombreuses instances, et au contraire, de généraliser davantage les valeurs pour lesquelles peu d'instances ont été trouvées. Cette procédure permet d'homogénéiser la cardinalité de chaque valeur d'attribut généralisée.

DBMINER : les développements ultérieurs de DBLEARN mènent à un nouveau système nommé DBMINER [38], avec les dispositifs suivants :

- de nouveaux genres d'extraction de règles à partir de grandes bases de données, y compris des règles d'association de multiple-niveau, des règles de classification, des règles de description;
- la génération et amélioration automatique des hiérarchies de concept;
- l'utilisation d'un niveau élevé du langage d'interrogation SQL avec des interfaces d'extraction de données graphiques;
- des améliorations dans l'architecture client/serveur et l'exécution pour de grandes applications.

Ces deux systèmes d'extraction de données, ont été développés pour l'exploitation interactive de la connaissance de multiple niveaux dans les grands volumes de bases de données relationnelles. Leur objectif est de découvrir des règles caractéristiques ou discriminantes dans des processus d'apprentissage supervisé. Les éléments fondamentaux de ces systèmes sont un ensemble étendu de fonctions d'extraction de données, incluant généralisation, caractérisation, association, classification, et prévision.

Résumés linguistiques

Comme leur nom l'indique les résumés linguistiques se basent sur l'exploitation des variables linguistiques introduites par L. Zadeh [105]. Les résumés linguistiques ont l'avantage d'être formulés dans un langage très proche de celui de l'utilisateur. Ils permettent aussi de proposer des descriptions intelligibles à des niveaux élevés de granularité. Nous classons ces résumés selon trois catégories : les résumés quantifiés, ainsi que leur extension dans les résumés par calcul de cardinalité floue et les résumés à base de règles floues, nous détaillons ci-dessous ces trois catégories.

Les résumés quantifiés.

La notion de *résumé quantifié* a été proposée au début des années 80 par R. R. Yager et C. Rubinson dans [103]. Leur approche consiste à formuler des propositions quantifiées et à considérer leur validité en regard d'une distribution de données exprimées selon le formalisme conventionnel *attribut-valeur*. De multiples travaux se sont intéressés aux résumés quantifiés [9, 12, 23, 46, 81]. Nous présentons ici deux langages d'interrogation des résumés quantifiés, *SUMMARYSQL* et *FQUERY*.

SUMMARYSQL. Proposé par [85], SUMMARYSQL est un langage de requête flexible qui est considéré comme une extension de SQL aux résumés linguistiques. Ce langage se base sur l'interprétation des résumés linguistiques en tant que prédicats flous construit sur une relation R. L'énoncé *"la plupart des n-uplets de la relation R ont un revenu élevé"* peut être exprimé comme suit :

$$\sum plupart(t \in R/\ t.Revenu \approx élevé)$$

où $\sum plupart$ représente le résumé validé par le quantificateur flou *la plupart*, et l'expression ($t.Revenu \approx élevé$) traduit le concept vague que doivent satisfaire les n-uplets t de la relation R.

SUMMARYSQL propose une clause *summary*, inspirée de la formulation des résumés linguistiques :

summary quantificateur flou
from relations
where conditions.

A titre d'exemple, en déterminant les individus qui ont un `revenu élevé` de la relation `employés`, on évalue le résumé :

summary la plupart
from employés
where revenu est élevé.

Le résultat de la requête est une valeur réelle correspondant à un degré de validité du résumé, traduisant ainsi dans quelle mesure le revenu de la plupart des employés est élevé.

FQUERY. Les auteurs de [46, 107], s'appuient sur le module FQUERY pour proposer une méthode de validation interactive des résumés linguistiques quantifiés. Basé sur des éléments issus de la théorie des sous-ensembles flous comme les concepts vagues, les relations floues ou les modificateurs flous, FQUERY étend les capacités d'un SGBD aux requêtes flexibles. Il offre une interface conviviale sur le modèle des *requêtes par l'exemple*, permettant à l'utilisateur de sélectionner, pour une requête, le quantificateur flou et les étiquettes linguistiques mis en jeu dans l'énoncé. Il exhibe

aussi d'autres résumés pertinents pouvant être obtenus à partir des attributs mis en jeu dans la requête. Ceci permet à l'utilisateur d'avoir tous les énoncés intéressants construits sur les attributs qu'il a sélectionné.

Les résumés par calcul de cardinalité floue.

Dans [12, 81], les auteurs proposent de réduire le cardinal d'une relation R d'une base de données en construisant une nouvelle relation R^* dont les n-uplets sont des résumés polymorphiques [4] de collections de n-uplets de R. Afin de construire cette relation R^*, deux étapes sont indispensables : l'étiquetage et la fusion. Une fois la relation R^* construite, les auteurs proposent d'évaluer des résumés linguistiques quantifiés. La phase d'étiquetage consiste à associer un n-uplet élémentaire t de la relation R à une ou plusieurs classes étiquettes floues pour générer autant de n-uplets qu'il existe de combinaisons d'étiquettes floues auxquelles t appartient au moins partiellement. Après vient la phase de fusion qui est chargée de réduire le nombre de n-uplets étiquetés, par calcul de cardinalités floues associées à chaque combinaison d'étiquettes observée dans R^*. Les résumés linguistiques construits à partir de la fusion des n-uplets de la nouvelle relation R^* sont exprimés sous la forme *Q n-uplets a de R sont b et c*, avec Q un quantificateur flou comme *la plupart* ou *peu de*. Cette approche est considérée comme l'extension de l'approche des résumés quantifiés. Elle s'apparente à une approche d'induction par attribut telle qu'elle a été proposée dans le système DBMINER [38], en ne considérant qu'un seul niveau de généralisation.

Les résumés à base de règles floues.

Dans le domaine des bases de données, les règles floues existent en tant que dépendances fonctionnelles étendues [11, 21, 82] et représentent des contraintes d'intégrité vérifiées par les données de la base. Pour les résumés linguistiques, les règles floues sont utilisées comme un support pour l'élaboration d'énoncés en langage naturel qui résument les informations contenues dans une base de données. Ces règles floues peuvent être découvertes à partir de données telle que la découverte des règles d'association dans une démarche constructive. Une deuxième démarche appelée déclarative consiste à valider des règles floues à partir d'énoncés exprimés dans un langage déclaratif et évalués par rapport à une collection de données.

Des travaux ont été proposés par P. Bosc et al. [12] pour la construction de résumés linguistiques basés sur la découverte de règles graduelles comme `plus les employés sont âgés, plus leur salaire est élevé`. Une autre approche proposée par [85] pour la validation de résumés linguistiques exprimés sous forme de dépendances fonctionnelles floues prolonge les travaux sur le langage des requêtes SUMMARYSQL détaillé précédemment. Cette approche présente les résumés linguistiques sous la forme `plus les employés sont âgés et haut placés, mieux ils sont`

[4] En informatique, le polymorphisme est l'idée d'autoriser le même code à être utilisé avec différents types, ce qui permet des implémentations plus abstraites et générales.

`payés`. D'une manière différente des résumés linguistiques, les dépendances fonctionnelles floues ont été exploitées par [20] pour proposer une méthode de réduction du nombre de n-uplets d'une relation R. Cette méthode est basée sur la définition d'opérateurs algébriques étendus qui conservent les propriétés des opérateurs de l'algèbre relationnelle.

Dans les sections précédentes, nous avons fait un tour d'horizon des travaux de recherche qui traitent du sujet de la compression de données. Parmi ces approches nous avons présenté les méthodes statistiques ainsi que les approches basées sur les modèles dont les résumés linguistiques.

Les approches statistiques sont dotées d'outils très puissants pour analyser et pouvoir déduire des conclusions sur un grand nombre de données. Les notions de probabilité et des statistiques ont l'avantage d'être intuitivement compréhensibles pour un grand nombre d'utilisateurs. Ces approches résument les données d'une façon très concise, ce qui limite de telles approches dans un processus de compression sémantique pour un but de découvrir des connaissances. Spécialement les méthodes de calculs d'agrégats et les moteurs OLAP, montrent bien cette limite. La fixation des intervalles au préalable permet de résumer une partie des données sélectionnées dans la base.

D'un autre côté, les algorithmes basés sur les modèles ou sur les méta-données s'inscrivent dans une approche de fouille de données tout en permettant la réduction des données volumineuses. Les méthodes d'apprentissage présentées dans ce chapitre, cherchent à fournir une représentation en intention d'un ensemble d'objets. Les notions de *fascicles* et de *ItCompress*, ont comme but de réduire l'espace de stockage des données, en ayant la contrainte de respecter leur sémantique. Les méthodes par induction et des résumés linguistiques, prennent en compte des connaissances de domaines, ainsi que des variables linguistiques très proche du langage humain.

Le but général de ces différentes approches présentées jusqu'ici est de synthétiser et de décrire globalement les données. Nous consacrons la section suivante à présenter un système de compression sémantique de données basé sur la génération de résumés linguistiques et s'appuyant sur la théorie des sous-ensembles flous. Ce système est appelé SaintEtiQ.

3.3 SaintEtiQ

Nous présentons dans cette section une vue d'ensemble du système de génération de résumés SaintEtiQ proposé par G. Raschia dans [83]. Ce système a pour but de construire, à partir d'une relation définie sur une base de données, une hiérarchie de résumés représentant cette relation. Il utilise de nombreuses notions issues de la théorie des sous-ensembles flous, qui permet notamment de représenter des informations imprécises et incertaines. Seuls les points que nous estimons intéressants pour la suite de ce document sont détaillés dans cette section. Pour plus d'information le

lecteur est invité à consulter les travaux décrits dans [83, 84]. Les élements essentiels à la compréhension des principes de la théorie des sous-ensembles flous sont fournis en annexe de ce document.

SaintEtiQ offre un algorithme de génération de résumés à partir d'une base de données relationnelle qu'on notera R. Chaque élément de cette base est considéré comme un enregistrement ou un n-uplet. On notera $\mathcal{A} = \langle A_1, A_2 \ldots, A_n \rangle$ de cardinalité n l'ensemble des attributs de la relation R, tels l'*âge*, le *revenu*, ou bien le *pays*. Un n-uplet t est donc noté $:t = \langle t.A_1, t.A_2, \ldots, t.A_n \rangle$, où chaque valeur d'attribut $t.A$ avec $A \in \mathcal{A}$, est un élément de D_A, où D_A est le domaine d'un attribut A. Les conventions d'écriture sont détaillées en annexe (notations) de ce document.

Les connaissances de domaine. L'utilité des connaissances de domaine, notée BK pour *Background Knowledge*, est de permettre l'*intelligibilité* des résumés. Le BK est utilisé pour réécrire les valeurs de n-uplets de la base de données en un langage spécifié par l'utilisateur. Pour ce faire, le modèle SaintEtiQ identifie pour chaque attribut les variables linguistiques issues de BK, qui sont proches de la description du n-uplet. Ainsi, chaque enregistrement de la base de données est entièrement réécrit sous forme de plusieurs n-uplets d'étiquettes floues.

Variable linguistique. Les variables linguistiques introduites par Zadeh en 1975 [105], permettent la description d'une variable en fonction d'un ensemble de caractérisations floues. Formellement, une variable linguistique est définie de la manière suivante :

Definition 3.5 (Variable linguistique)**.** *Une variable linguistique est représentée par un triplet $(V, \Omega, \mathcal{L}_v)$ où V est une variable (par exemple le revenu, l'âge, ...) définie sur un ensemble de référence $\Omega(\mathbb{R}, \mathbb{N}, \ldots)$ et dont la valeur peut être n'importe quel élément de Ω. On note $\mathcal{L}_v = \{X, Y, \ldots\}$ un ensemble, fini ou non, de sous-ensembles flous de Ω, restrictions de valeurs de V dans Ω, qui sont utilisés pour caractériser V. Un élément de $\mathcal{L}_v$ est aussi appelé étiquette linguistique.*

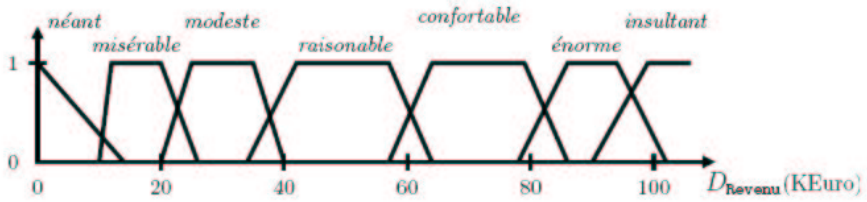


Figure 3.3 – Variable linguistique définie sur le domaine de l'attribut REVENU

Notons que dans le système SaintEtiQ, les étiquettes linguistiques doivent couvrir l'ensemble du domaine Ω, ce qui correspond à une propriété des partitions floues.

La figure 3.3 décrit l'attribut REVENU avec les termes *misérable*, *modeste*, *confortable* ....

3.3.1 L'architecture du système

La figure 3.4 tirée de [90], donne une vue d'ensemble de l'architecture du système SAINTETIQ. L'opération de résumé est un processus de découverte de connaissances à partir des données. Proposant une phase de *pré-traitement* et une phase de fouille de données (classification conceptuelle). La phase de pré-traitement permet au système de réécrire les enregistrements de la base de données avant que ceux-ci ne soient exploités par le processus d'extraction. Cette étape donne naissance à un ou plusieurs n-uplets candidats qui sont différentes représentations, fonction des connaissances définies sur le domaine, d'un même enregistrement de la base de données. La seconde phase de fouille de données prend en considération les enregistrements de la base, un à un, et leur applique un algorithme d'apprentissage dans l'objectif de produire des résumés.

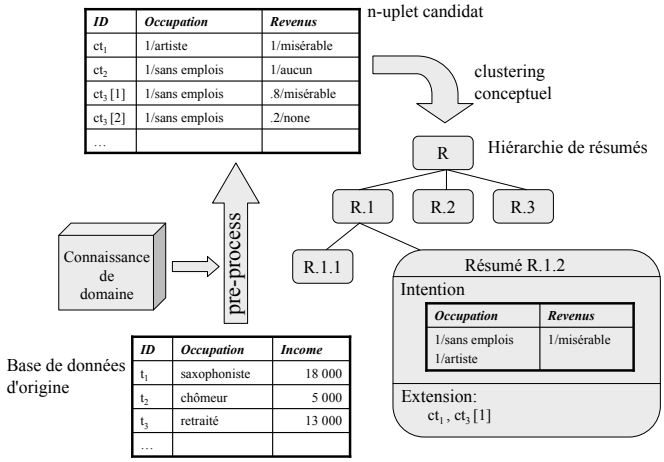


Figure 3.4 – Architecture du Système SAINTETIQ

Une version étendue du système SAINTETIQ a été proposée par R. SAINT-PAUL et al. dans [90], afin de répondre aux exigences de passage à l'echelle et de mises à jour incrémentales de la base de données résumée. Nombre d'applications ont été testées sur ce système, notamment dans le domaine bancaire [93] et le domaine de la recherche d'information dans les bases d'images [92].

Service de réécriture. L'opération de réécriture consiste à transformer les données initiales de la base, qui sont sous la forme de n-uplets $t = \langle t.A_1, \ldots, t.A_k \rangle$ où chaque $t.A_i$ est défini sur son domaine D_{A_i}. Pour chaque attribut A de $\mathcal{A}$, l'opération de réécriture consiste à produire un sous-ensemble flou de D_A^+ [5]. La réécriture d'une

[5] Le domaine réécrit de l'attribut A.

valeur de l'attribut A est donc donnée par un ensemble flou dont les membres sont les étiquettes linguistiques L de D_A^+ affectées chacune de son degré d'appartenance.

Exemple 3.2 (Réécriture). *Soit l'enregistrement présenté ci-dessous, extrait d'une table relationnelle de [83]:*

$$\langle \textit{Nom(id), Catégorie socio-professionnelle, Age, Revenu} \rangle$$

$$\langle \textit{Burns, patron de centrale nucléaire, 45 ans, 87 000 euros} \rangle$$

Considérant que REVENU $Burns.Revenu = 87000$ *euros, sa réécriture sur les différents attributs, selon les variables définies dans [83], page 81, nous donne :*

$$\begin{aligned} Burns.CSP &= \{0.9/\textit{homme d'affaires} + 1.0/\textit{chef d'entreprise}\} \\ Burns.Age &= \{1.0/\textit{adulte}\} \\ Burns.Revenu &= \{1.0/\textit{enorme}\} \end{aligned}$$

On obtient ainsi deux tuples candidats notés ct:

$$\begin{aligned} ct_1 &= \langle Burns\,[1]\,, \textit{homme d'affaires}, \textit{adulte}, \textit{énorme} \rangle \\ ct_2 &= \langle Burns\,[2]\,, \textit{chef d'entreprise}, \textit{adulte}, \textit{énorme} \rangle \end{aligned}$$

Service résumé. Dans l'architecture du système SAINTETIQ proposée par R. SaintPAUL et al. [90], le service de résumés qui est proposé en tant que service web est activé par une simple méthode qui prend en paramètre le document contenant le ou les n-uplets dans leur forme préparée. Le traitement commence par la création d'un nœud (résumé) racine d'une hiérarchie qui contient tous les n-uplets candidats dans leur forme réécrite, la suite de la construction de l'arbre des résumés se fait, selon un algorithme d'apprentissage que nous ne détaillons pas ici (voir [83]), et à l'aide de quelques opérateurs que nous définissons ci dessous.

Les opérateurs d'apprentissage. Le processus SAINTETIQ se base sur un algorithme d'apprentissage incrémental. Il construit et modifie une hiérarchie de résumés par l'insertion séquentielle des n-uplets candidats. Les n-uplets candidats sont incorporés un à un dans l'arbre, par la racine, en utilisant quatre opérateurs d'apprentissage. Ces opérateurs au nombre de quatre, sont les suivants :

1. *initialiser :* cet opérateur permet de créer un nouveau résumé ne contenant que le n-uplet candidat *ct* qu'on cherche à incorporer dans l'arbre des résumés.
2. *affecter :* cet opérateur permet d'affecter le n-uplet candidat *ct* à l'un des résumés déjà existants.
3. *fusionner :* cet opérateur permet de fusionner deux résumés z_1 et z_2 en un résumé z_{merge}, les résumés z_1 et z_2 devenant les fils de z_{merge}. Le tuple candidat *ct* est alors affecté au résultat z_{merge} de cette fusion. La figure 3.5 illustre ceci.

4. *éclater :* connaissant un résumé z_i, son père z_0 et l'ensemble de ses fils $z_{i1}, \ldots, z_{in}$, cet opérateur fait disparaître z_i pour faire remonter ses fils dans la hiérarchie, en les rattachant au résumé z_0. Ce procédé est illustré sur la figure 3.6.

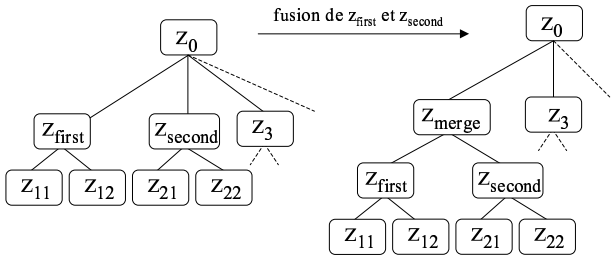


Figure 3.5 – Opérateur de fusion

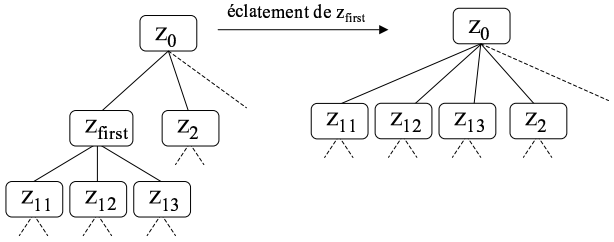


Figure 3.6 – Opérateur d'éclatement

3.4 Conclusion

Nous avons présenté dans ce chapitre les propositions qui existent en matière de résumé de données, ou plus largement de compression sémantique des informations contenues dans de larges volumes de données. Ce tour d'horizon nous a permis de positionner le système SaintEtiQ dans l'ensemble de ces propositions. Dans la deuxième partie de ce document, nous nous intéressons de près à SaintEtiQ comme une approche générant des résumés flous auxquels nous apporterons notre contribution.

Conclusion

Cette partie nous a permis de présenter deux domaines de recherche auxquels nous nous sommes intéressés dans cette thèse. L'objectif commun de ces domaines est de proposer à l'utilisateur une représentation complète des données, à un niveau d'abstraction supérieur, en réduisant leur volume. Cet objectif est motivé par le besoin des décideurs qui utilisent la présentation produite pour analyser un sujet précis à l'aide d'outils dédiés.

Dans un premier temps, nous nous sommes focalisés sur les systèmes d'information décisionnels et sur les avantages que propose l'analyse en ligne des données. Nous avons pu réaliser que la mise en place de tels systèmes s'avère efficace pour gérer une masse de données de plus en plus conséquente, et le stockage des données dans un entrepôt constitue un support effectif pour l'analyse des données. Cette analyse se fait par des systèmes OLAP dont la vocation est de fournir à l'utilisateur un outil visuel pour explorer et naviguer dans les données d'un cube afin d'y découvrir rapidement des connaissances. Une synthèse des opérateurs algébriques de manipulation des données dans les systèmes décisionnels a été présentée.

Dans un deuxième temps, le second chapitre a été l'occasion de présenter les différentes approches qui traitent de la problématique de la compression sémantique. Parmi ces approches, nous allons plus particulièrement concentré notre étude sur le système SaintEtiQ. Celui-ci consiste à construire une collection de résumés organisés hiérarchiquement du plus général aux plus spécifiques. Chaque résumé fournit une représentation concise d'un ensemble des n-uplets de la base de données résumée, par le biais d'un sous-ensemble flou de descripteurs sur chaque attribut. L'enjeu du résumé est la synthèse de l'information se trouvant au sein des faits individuels pour en fournir une représentation concise.

La synthèse de ces deux chapitres nous permet de faire ici un rapprochement entre un système décisionnel et SaintEtiQ. Nous pensons que ces deux approches tentent de résoudre une double problématique commune : d'une part elles permettent de réduire le volume d'une masse de données et d'autre part, elles offrent à l'utilisateur un outil tangible d'aide à la décision.

SaintEtiQ : un système décisionnel

Nous estimons que SaintEtiQ peut tout à fait s'inscrire dans un processus d'aide à la décision. Le lien que nous faisons entre SaintEtiQ et les systèmes décisionnels est motivé par les similarités existantes entre les approches. Tout d'abord, l'objectif des deux approches est de fournir à l'utilisateur une vue synthétique sur les données à un niveau d'abstraction. La notion de "résumabilité [6]" a été prise en considération

[6] capacité de synthèse.

dans les systèmes OLAP par le calcul d'agrégats. Les deux approches mettent en avant leur aptitude à traiter d'importants volumes de données. Le deuxième point commun que nous avons identifié est celui du traitement réservé aux données prises en compte par chacune des deux approches. En effet, les données sont préalablement nettoyées grâce aux *ETL* dans les systèmes décisionnels et grâce à la réécriture dans SaintEtiQ. Les connaissances de domaine utilisée dans la phase de réécriture fournissent sur les attributs considérés par SaintEtiQ une grille de lecture des données à l'aide des descripteurs linguistiques, il en va de même pour les modalités retenues sur chaque dimension d'un cube. Par analogie, un attribut correspond donc à la notion de dimension dans un cube de données.

Notre idée première vis-à-vis des systèmes décisionnels est de trouver les points de convergence entre les systèmes OLAP, qui fournissent une vue résumée des données sous forme de cubes, et les résumés construits par SaintEtiQ. Afin de compléter cette comparaison, nous souhaitons attirer l'attention du lecteur sur les points suivants :

- Les cellules d'un cube de données contiennent des valeurs agrégées pré-calculées selon plusieurs dimensions. Cependant, ces cellules sont définies selon des intervalles fixes de valeurs. Parallèlement, les résumés utilisent des concepts flous avec différents degrés de satisfaction.
- Une mesure utilisée dans un cube de données est définie comme une fonction statistique et fournit une information quantitative sur les données. Cette information est considérée comme une réponse effective à une requête donnée, mais elle est incapable de fournir une réponse descriptive. Les résumés de SaintEtiQ fournissent une information qualitative en proposant une description en intention et singulièrement des degrés de satisfaction à des prédicats de requêtes.
- L'un des challenges de la construction des cubes de données est d'éviter la génération de cellules vides. Ce problème devient sérieux quand il s'agit d'espace à grande dimension qui, en fonction de la distribution des données, donne lieu à des espaces creux. Ce problème est inexistant dans le processus de génération de résumés, dans la mesure où chaque résumé est crée en observant les données.
- Les données qui sont agrégées et souvent stockées dans des vues matérialisées d'un entrepôt de données concernent un sujet prédéfini à analyser. Cette prédétermination affecte le choix des dimensions et des mesures pour la construction du cube. Cette orientation du sujet peut limiter les analyses du décideur et l'influencer dans ses conclusions. Du côté de SaintEtiQ, la description multidimensionnelle que fournissent les résumés offre une grande flexibilité dans l'interprétation et l'analyse. Le processus SaintEtiQ construit les résumés sans aucune hypothèse quant à l'utilisation qui sera faite de la structure produite. L'ubiquité de la vision synthétique que proposent les résumés est donc à opposer à la pré-détermination que sous-tend la mise en place des outils de type OLAP.

Les résumés des données agrégées dans les systèmes OLAP sont plus souvent des données issues d'indicateurs statistiques ou d'agrégats comme les sommes, les

compteurs ou les ratios. Ils sont calculés selon plusieurs axes et à différents niveaux de granularité. L'avantage majeur des systèmes OLAP réside dans la structure de stockage sur laquelle ils s'appuient leur permettant de pré-calculer les valeurs d'agrégats. Le second grand avantage des moteurs OLAP est l'ensemble des algorithmes qu'ils proposent pour la mise à jour, autant que possible incrémentale, des cellules d'un cube de données. Ceci permet de conserver un grand degré de fraîcheur des données.

Les avantages que les systèmes OLAP proposent sont conformes avec les objectifs des résumés, issus de SaintEtiQ. Ces avantages concernent l'interprétation, l'intelligibilité, la présentation à différents niveaux d'abstraction et la vision synthétique des données. En effet, l'un des points forts de SaintEtiQ est de présenter les données sous une forme intelligible, dans un vocabulaire proche du langage de l'utilisateur. Le résumé peut être considéré comme un outil d'analyse directement utilisable en ce qu'il fournit une représentation synthétique des données qui peut apporter une connaissance non triviale sur les données.

La maintenance incrémentale des résumés s'impose pour garantir un bon niveau de fraîcheur. La prise en compte incrémentale des opérations d'insertion, modification et suppression des n-uplets de la base de données résumée permet d'assurer un maintien permanent de la cohérance de la hiérarchie de résumés. La relation d'ordre définie sur les résumés possède une équivalence en terme d'union sur les ensembles d'individus. En ce sens, chaque résumé parent réalise un groupement d'individus sur des critères qui ne sont pas fixés mais au contraire évalués dynamiquement en fonction de la distribution. En s'appuyant sur la théorie des sous-ensembles flous, les descripteurs utilisés par SaintEtiQ introduisent en outre une souplesse dans la définition des frontières des intervalles limitant ainsi l'effet de seuil.

Le maintien en ligne de la consistance du résumé vis-à-vis de la base de données, positionne SaintEtiQ dans la lignée des systèmes OLAP. En comparant aux techniques d'analyse en ligne ou OLAP, et étant donné le volume de données produites par SaintEtiQ dans la hiérarchie, un besoin primordial pour SaintEtiQ serait d'enrichir les modes d'exploration de la structure produite. La structure du cube peut être comparée à une coupe de la hiérarchie des résumés. La définition d'une structure qui peut être un support d'analyse en ligne, s'inscrit bien dans une volonté de créer pour les résumés de SaintEtiQ un parallèle avec les méthodes d'analyse en ligne que proposent les systèmes OLAP. Pour atteindre cet objectif il paraît essentiel de mettre au point un modèle multidimensionnel de résumés et une algèbre pour les manipuler qui serait le pendant de celle des cubes de données.

Suite à cette brève discussion, nous allons nous concentrer sur le principal avantage des systèmes OLAP, c'est-à-dire l'analyse exploratoire des données. En effet, ces systèmes mettent à la disposition des décideurs, une interface graphique qui permet à l'utilisateur de naviguer et de manipuler les données. Cette phase d'analyse en ligne des données, constitue la réponse au besoin que nous avons soulevé pour faire évoluer le processus SaintEtiQ vers un système décisionnel, ce besoin réside dans la partie concernant l'exploration des résumés produits par SaintEtiQ.

Pour répondre à ce besoin, notre proposition, qui constitue la contribution de

cette thèse, se décline en trois points :

- Enrichir la hiérarchie en proposant un modèle multidimensionnel formel sur lequel pourra s'appuyer une analyse en ligne.
- Définir une algèbre de manipulation en ligne des résumés.
- Offrir une représentation qui facilite l'interprétation des résumés et aide l'utilisateur pour la prise de décision.

La deuxième partie de ce document, est entièrement consacrée au détail de notre proposition.

Partie II

Vers un processus d'analyse en ligne de résumés flous

Introduction

La présente partie de nos travaux, est consacrée à la contribution que nous apporterons au système SaintEtiQ. Elle consiste en la proposition d'un modèle, orienté utilisateur, pour supporter l'analyse en ligne des résumés générés par SaintEtiQ. Afin d'atteindre cet objectif, trois chapitres composent cette partie.

Le premier chapitre concerne la partie modélisation. Il propose un modèle multidimensionnel pour les résumés de données flous. Par analogie aux cubes de données des systèmes OLAP, vu dans la première partie, ce modèle se base sur une structure appelée *partition de résumés*, correspondant à un niveau d'abstraction de la hiérarchie des résumés de SaintEtiQ.

Dans le second chapitre, en s'inspirant de l'algèbre de manipulation des cubes OLAP, nous définissons un ensemble d'opérations algébriques pour manipuler les résumés linguistiques du système SaintEtiQ. Les caractéristiques floues de ces résumés, font l'objet de l'adaptation d'une algèbre d'analyse en ligne aux résumés de SaintEtiQ, et aux différentes partitions de résumés extraites d'une hiérarchie.

Nous nous intéressons dans le troisième chapitre, aux prototypes flous et à leur utilisation dans les résumés linguistiques. Ceci dans le but de trouver une meilleure représentation intelligible pour le contenu d'un résumé issu d'une partition qui correspond à une vue de la hiérarchie de SaintEtiQ.

Chapitre 4

Un modèle multidimensionnel pour les résumés de données

Reconsidérer, chercher une justification pour une décision déjà prise.

— Le Dictionnaire du Diable, 1911.

On a abordé dans le précédent chapitre la hiérarchie générée par le processus SaintEtiQ. SaintEtiQ est un système basé sur un processus de classification conceptuelle qui construit en résultat une hiérarchie de résumés contenant l'ensemble de la base originale. Son premier objectif est la synthèse de données volumineuses sous formes de résumés. Or, les résumés que génèrent SaintEtiQ restent très nombreux pour un utilisateur qui souhaite les exploiter et les analyser.

Dans ce chapitre, nous offrons à la hiérarchie produite par SaintEtiQ une nouvelle organisation et une présentation plus proche d'un utilisateur ordinaire. Pour ceci, nous définissons un modèle basé sur l'extraction de différentes coupes de la hiérarchie appelées *partitions de résumés*. Ces partitions forment la base d'un modèle multidimensionnel et fournissent des vues synthétiques sur l'ensemble des données originales.

4.1 Rappel des objectifs

Nous avons étudié dans le premier chapitre de ce document, l'architecture d'un système décisionnel. Nous avons retenu l'existence de quatre grands modules qui constituent la chaîne d'un processus d'aide à la décision. Il s'agit de l'*intégration* qui consiste à nettoyer les données source et les transformer, de la *construction* de l'entrepôt de données, de la *réorganisation* de cet entrepôt en magasins de données et enfin de l'*interrogation* qui consiste à manipuler les cubes de données et à les représenter à l'utilisateur.

L'un des objectifs fixés dans ce chapitre, est d'adapter le système SaintEtiQ à l'architecture type d'un processus d'aide à la décision ou d'un système décisionnel (voir chapitre 2). La figure 4.1 propose l'architecture vers laquelle nous souhaitons

conduire le système SaintEtiQ. Dans cette architecture, la hiérarchie est considérée comme un "entrepôt de résumés". L'intérêt du modèle proposé dans ce chapitre est de réorganiser la hiérarchie des résumés dans le but de faciliter pour l'utilisateur l'accès et l'analyse des données se trouvant dans ces résumés. Une structure multidimensionnelle est proposée afin de modéliser chaque ensemble de résumés extraits à partir de cette hiérarchie. La collection de résumés extraite doit fournir une vue synthétique par rapport aux données de la base originale et doit être une structure sur laquelle pourrait se baser l'utilisateur pour l'analyse et la prise de décision. Par la suite nous allons manipuler ces vues à l'aide d'un ensemble d'opérateurs.

Le modèle que nous proposons dans ce chapitre doit permettre l'application d'une analyse en ligne des résumés en prenant en considération leur aspect flou. Sur la figure 4.1, l'architecture décisionnelle du système SaintEtiQ regroupe différentes phases, présentées ci-dessous.

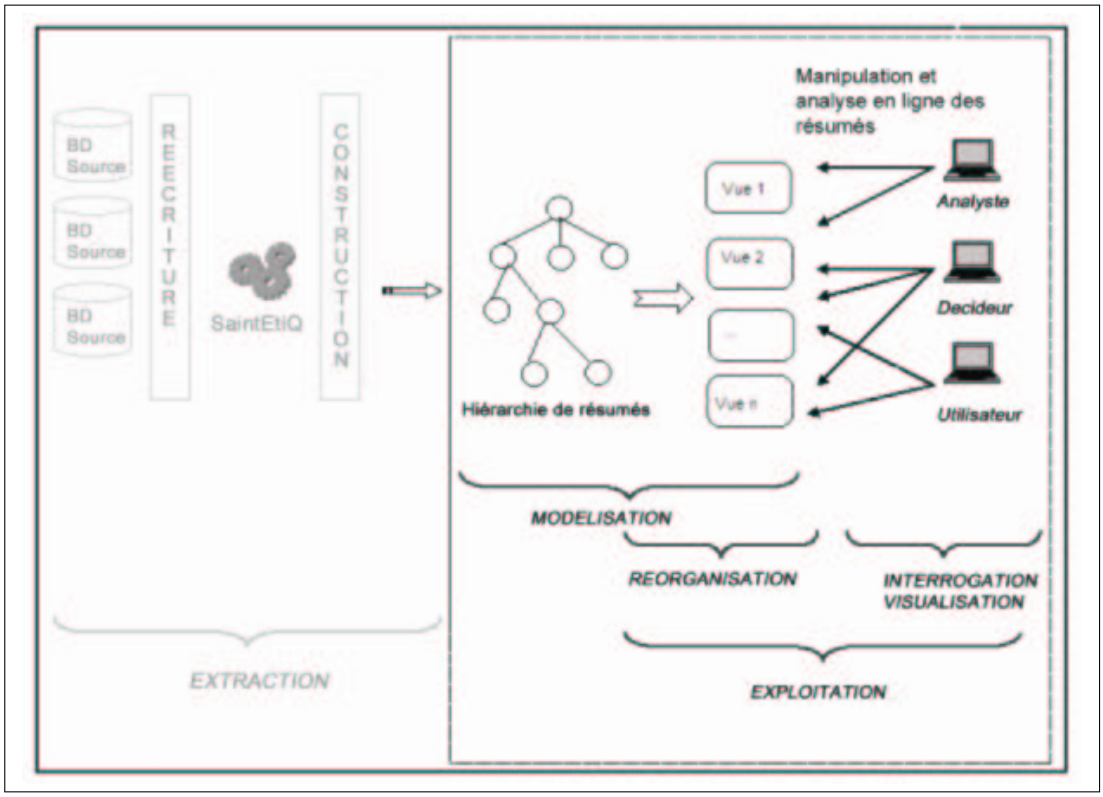


Figure 4.1 – Vers une architecture décisionnelle de SaintEtiQ

La première partie contient les phases d'*ETL*[1] et de construction. Dans l'architecture décisionnelle du système SaintEtiQ, l'outil *ETL* se trouve à l'étape de réécriture. En effet, la réécriture permet un premier niveau d'abstraction qui présente également l'avantage d'homogénéiser la représentation des données. Ce processus

[1]Extraction, Transformation et Loading

consiste en la transformation des données originales en données réécrites, au moyen de connaissances de domaines et d'un ensemble de descripteurs linguistiques. Les détails de cette phase se trouvent dans la section 3.3.1 du chapitre précédent. La phase de construction de la hiérarchie, par analogie, correspond à la construction d'un "entrepôt de résumés". Cette phase utilise pour la construction de la hiérarchie un algorithme de généralisation par formation de concepts flous. Elle intervient après l'abstraction contenue dans la phase de réécriture.

La deuxième partie correspond dans notre approche à la modélisation et à la réorganisation des résumés selon le modèle multidimensionnel proposé dans ce chapitre, ainsi qu'à la manipulation de ces structures multidimensionnelles détaillée dans les chapitres suivants. La modélisation a comme objectif de rendre les résumés de données flous plus intelligibles et plus proches d'un décideur ou d'un analyste non expert. Cette modélisation est la phase intermédiaire entre les résumés générés par SaintEtiQ et la phase d'exploitation. Elle permet d'avoir une structure multidimensionnelle qui facilitera les manipulations dans un processus d'analyse en ligne. Ces manipulations sont basées sur l'organisation hiérarchique et la granularité des résumés. En premier lieu, la phase de réorganisation consiste à présenter la hiérarchie proposée par SaintEtiQ sous une forme plus simplifiée ce qui facilitera son exploration ainsi que l'accès aux données se trouvant dans les résumés à un utilisateur lambda. Et en second lieu, l'étape de manipulation contient l'adaptation des opérateurs OLAP au modèle proposé. Ces opérateurs vont permettre d'interroger les résumés de données et de visualiser et analyser les données à différents niveaux d'abstraction de la hiérarchie résultante. Cette partie sera détaillée dans le chapitre suivant de ce document.

4.2 Le modèle de résumé

Nous abordons dans cette section, les différentes caractéristiques du modèle de résumé de données tel que défini par le modèle SaintEtiQ.

Un résumé z est défini par un triplet $z = (I_z, R_z, E_z)$[2] de l'espace $R^* \times R \times \xi$. Les éléments de ce triplet représentent, pour I_z, l'intention du résumé z, pour R_z son extension et, pour E_z, un ensemble de relations typées existant entre z et d'autres résumés.

4.2.1 L'intention et l'extension du résumé

Dans SaintEtiQ, un résumé z est caractérisé par une définition en *intention*, qui donne une représentation du résumé suivant les descripteurs linguistiques qui auront précédemment été choisis, et une définition en *extension*, qui consiste en la description de l'ensemble des n-uplets composant le résumé.

Definition 4.1 (Intention du résumé.)**.** *L'intention I_z d'un résumé z est un élément*

[2] R est la relation d'origine à résumer.

du produit cartésien des sous-ensembles flous des domaines d'attribut réécrits.

$$I_z = \langle z.A_1, \ldots, z.A_k \rangle \,.\;\; z.A_i \in \mathcal{F}(D^+_{A_i}), 1 \leq i \leq n,$$

où D^+_A est l'ensemble des descripteurs définis sur l'attribut A.

Le calcul du coefficient associé à chacun des descripteurs linguistiques présentant un attribut A_i dans l'intention du résumé z, s'effectue en gardant le coefficient maximum exprimé par les n-uplets candidats qui constitueront z.

L'intention d'un résumé est donc sa description au moyen des étiquettes linguistiques définies dans le BK[3] qui a été utilisé pendant la phase de réécriture. L'espace de représentation de l'intention d'un résumé est le même que celui de représentation des n-uplets réécrits. Toutefois, dans le cadre des résumés, une mesure supplémentaire appelée support est également associée à chaque descripteur. Le support d'un descripteur représente le nombre de n-uplets du résumé pour lesquels ce descripteur possède un degré de satisfaction non nul. Lorsque l'intention n'est définie que par un descripteur, le support est donc égal au nombre de n-uplets contenus dans l'extension du résumé.

Definition 4.2 (Extension d'un résumé.)**.** *Soit z un résumé. L'extension R_z de z est l'ensemble des n-uplets candidats ct_i représentés par le résumé z.*

$$R_z = \{ct_1, ct_2, \ldots, ct_n\}$$

Notons qu'il est toujours possible, à partir de cette définition de retrouver les n-uplets de la relation R décrits par le résumé z. En effet, chaque n-uplet candidat est associé à l'enregistrement de la base de données qui l'a générée. Il est également possible d'exprimer R_z à partir de l'intention I_z comme l'ensemble de n-uplets tels que leur réécriture par ϕ possède un descripteur commun avec l'intention du résumé :

$$R_z = \{t \in R | \forall A \in \mathcal{A}, ||\phi(t.A) \cup_F z.A|| > 0\}$$

Exemple 4.3 (Intention et extension). *Soit le résumé z composé des n-uplets de la table 4.1 tirée de [83] . L'extension de z est simplement l'ensemble des n-uplets réécrits que résume z :*

$$R_z = \{Apu\,[1]\,, Apu\,[2]\,, Burns\,[1]\,, Burns\,[2]\,, Homer\,[1]\}$$

L'intention de z est décrite sur deux attributs CSP *et* REVENU*. On obtient donc :*
$z = \langle \{0.9/\textit{homme d'affaires} + 0.8/\textit{cadre supérieur} + 0.9/\textit{chef d'entreprise}\}\,, \{1.0/\textit{raisonnable} + 1.0/\textit{énorme}\}\rangle$

[3]Background knowledge.

Nom(id)	CSP	ϕ	Revenu	ϕ	Réf
Apu[1]	homme d'affaires	.7	raisonnable	.7	Apu
Apu[2]	chef d'entreprise	.6	raisonnable	.7	Apu
Burns[1]	homme d'affaires	.9	énorme	1.0	Burns
Burns[2]	chef d'entreprise	1.0	énorme	1.0	Burns
Homer[1]	cadre supérieur	.8	raisonnable	1.0	Homer

Table 4.1 – Extrait de la table réécrite PERSONNAGES-SIMPSONS

4.2.1.1 Les cardinalités du résumé

Comme illustré dans la table 4.1, nous savons qu'un enregistrement de la base de données peut être réécrit en un ou plusieurs n-uplets(s) candidats(s). Il est en effet possible d'avoir dans un même résumé plusieurs candidats d'un même n-uplet original. Ceci ne reflète toutefois pas le contenu réel du résumé puisque chaque n-uplet candidat participe avec un poids plus ou moins important dans la représentation des n-uplets de la base d'origine.

Nous avons vu que l'extension du résumé R_z contient les n-uplets candidats qui composent le résumé, mais la cardinalité de l'ensemble R_z augmente de 1 si seulement on incorpore un nouveau candidat dont l'identifiant n'est pas déjà présent dans le résumé. Le comptage des identifiants décrit par le résumé ne peut donc nous fournir une bonne idée sur sa représentativité vis-à-vis de la base de données de départ. On distinguera donc deux cardinalités différentes.

Definition 4.3 (Cardinalité absolue d'un résumé)**.** *La cardinalité absolue d'un résumé reflète le nombre d'instances de la base de données qui sont décrites dans ce résumé. Cette valeur est donnée par le cardinal $|Rz|$ de l'ensemble classique R_z. On notera également $|z|$ cette même valeur.*

Definition 4.4 (Cardinalité relative d'un résumé)**.** *La cardinalité relative d'un résumé reflète sa représentativité par rapport à la base de données de départ. Elle prend donc en compte non pas le nombre absolu de n-uplets de la base, mais le poids ω de chaque n-uplet candidat qui a été affecté au résumé.*

$$\textbf{\textit{card}}(R_z) = \sum_{ct \in z} \omega(ct)$$

avec, $\omega(ct) = \frac{1}{N(t)}$ où $N(t)$ est le nombre de n-uplets candidats générés par t. Naturellement, on a toujours $card(z) \leq |z|$ car $\forall ct, \omega(ct) \leq 1$.

Par exemple, si l'on reprend le tableau 4.1, on trouve $\omega(Apu[1]) = 1/2, \omega(Homer[1]) = 1$ et la cardinalité est $card(R_z) = 3$. De même on peut définir une cardinalité relative appliquée aux étiquettes linguistiques de l'intention du résumé, afin de pouvoir les comparer entre elles :

$$\textbf{card}_{z.A}(d) = \sum_{ct \in R_z | ct.A = d} \omega(ct).$$

Ainsi, $\textbf{card}_{z.revenu}(raisonnable) = 1/2 + 1/2 + 1 = 2$.

4.2.2 Relation sur les résumés.

Etant donnés deux résumés z et z', éléments du produit cartésien $Z = \Pi_{A \in \mathcal{A}}(\mathcal{F}_{\mathcal{A}}^{+})^4$, la relation $\preceq$ est définie sur $Z \times Z$ par :

$$z \preceq z' \Leftrightarrow R_z \subseteq R_{z'}$$

où la relation $\subseteq$ sur les extensions R_z *et* $R_{z'}$ respectivement des résumés z et z' est la relation d'inclusion sur les ensembles classiques.

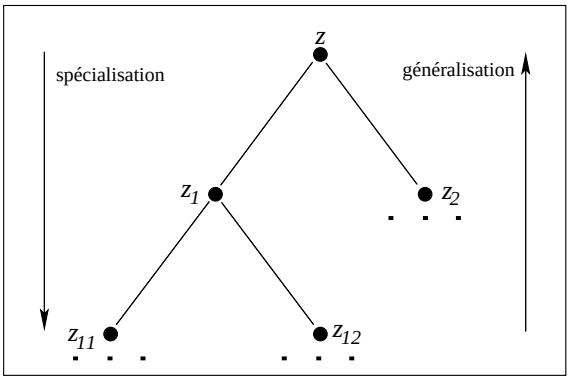


Figure 4.2 – Généralisation entre les résumés

D'un point de vue sémantique, la relation d'ordre $\preceq$ traduit un phénomène de généralisation des résumés les uns par rapport aux autres. Lorsque deux résumés z et z' sont liés par la relation $\preceq$ on dit que z' généralise z, de telle sorte que, s'ils sont comparables par la relation $\preceq$, les descripteurs de z' sont au moins aussi satisfaisants et représentés que ceux de z, ou bien ce sont de nouvelles étiquettes linguistiques représentant les caractéristiques des éléments de $R_{z'} - R_z$. La figure 4.2 schématise la relation de généralisation qui existe entre les résumés.

Soit z et z' deux résumés. Si z' généralise z ($z \preceq z'$) alors la description intentionnelle de z sur chaque attribut est incluse dans la description intentionnelle de z' :

$$z \preceq z' \Rightarrow \forall A \in \mathcal{A}, z.A \subseteq_F z'.A$$

où la relation $\subseteq_F$ sur les ensembles flous $z.A$ et $z'.A$ de $\mathcal{F}_A^+$ correspond à l'inclusion floue classique.

4.3 La hiérarchie de SaintEtiQ

Le système SaintEtiQ tel qu'il a été proposé dans [84] génère une hiérarchie de résumés représentant la base de données à différents niveaux de granularité. Plus on descend vers les feuilles, plus les résumés sont précis.

[4] l'ensemble des sous-ensembles flous construits sur $\mathcal{A}$.

Exemple 4.4 (Organisation hiérarchique des résumés de SAINTETIQ). *La figure 4.3 montre un exemple de l'organisation hiérarchique des résumés. Chaque nœud correspond à un résumé, le nœud racine contient tous les n-uplets candidats de la table relationnelle résumée.*

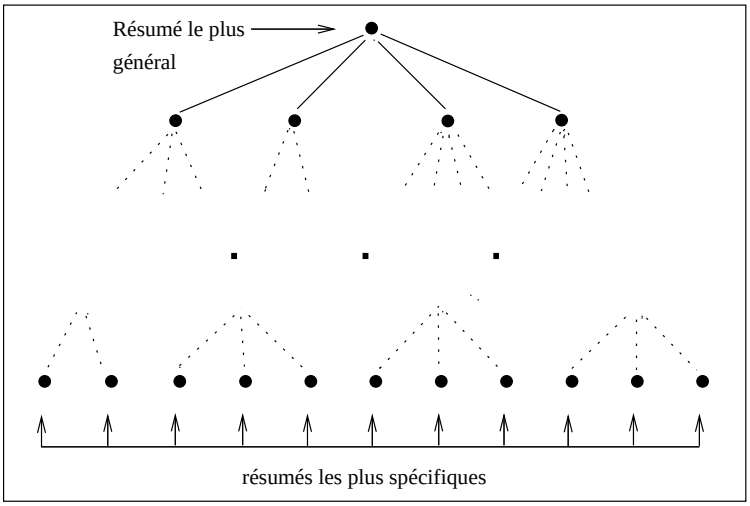


Figure 4.3 – Organisation hiérarchique des résumés SAINTETIQ

A chaque nœud correspond un résumé z. La racine de l'arbre, notée z_0, est un résumé de tous les enregistrements de la base de données. C'est évidemment le résumé le plus général que l'on puisse faire sur les données. Lorsque l'on descend dans les branches de l'arbre, les nœuds correspondent à des sous-ensembles de plus en plus réduits d'enregistrements de R, de telle sorte que les résumés sont de plus en plus spécialisés. Les résumés définis dans les feuilles de l'arbre déterminent donc la plus fine granularité disponible, en termes de capacité de synthèse, avant d'atteindre directement les enregistrements de la base. Identifier la structure arborescente des connaissances est équivalent à l'établissement d'un ordre partiel sur les résumés. Cette hiérarchie résultante d'un processus de construction à partir de la relation de base R sera notée, dans ce document, par H_R.

La hiérarchie proposée par SAINTETIQ reste encore volumineuse. La figure 4.4 montre les premiers niveaux d'un exemple de hiérarchie produite par SAINTETIQ sur un jeu de données de 33700 n-uplets d'origine définis sur dix attributs, qui a produit une hiérarchie de 27304 résumés dont 14766 feuilles. Toutefois, cette hiérarchie contient un grand nombre d'information dont a besoin un utilisateur final. Mais la forme finale de cette hiérarchie est encore assez difficile à exploiter pour l'utilisateur.

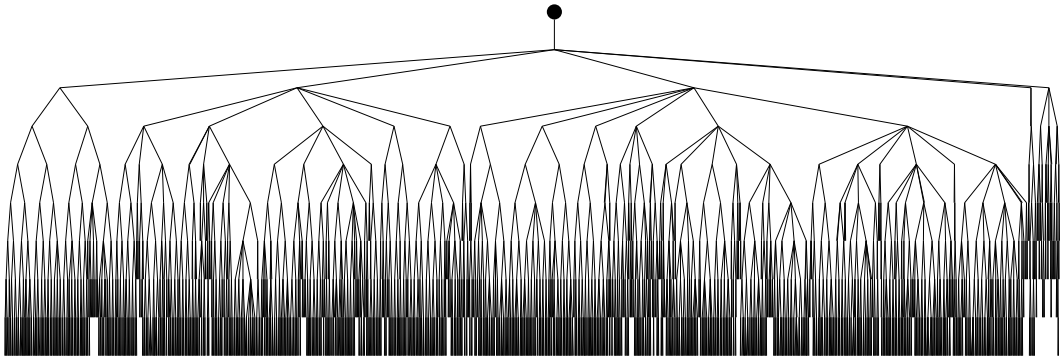


Figure 4.4 – Les premiers niveaux d'une hiérarchie résultante de SaintEtiQ

4.3.1 Quelques caractéristiques de la hiérarchie de résumés

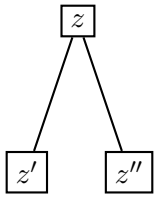
4.3.1.1 Relation d'ordre partiel sur les résumés

Nous avons déjà évoqué la relation d'ordre partiel qui existe sur les résumés de la hiérarchie générée par SaintEtiQ, dans la section 4.2 du chapitre 2.

Ainsi, à tout niveau de la hiérarchie, un nœud père doit généraliser chacun de ses fils. Les n-uplets candidats de l'extension d'un résumé sont aussi représentés dans l'extension de tous les résumés ancêtres de ce résumé, et ce jusqu'à la racine. Cette relation d'ordre partiel se traduit au niveau des résumés d'une hiérarchie, comme il est présenté dans l'exemple 4.5.

Exemple 4.5 (Relation d'ordre partiel sur les résumés). *Soient les résumés z, z', z'', tel que z généralise z' et z'' :*

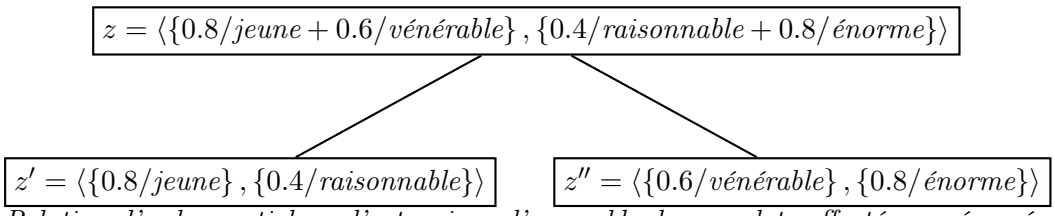
$$z' \preceq z \text{ et } z'' \preceq z$$



– *Relation d'ordre partiel sur l'intention : sur cet exemple, les trois résumés sont décrits intentionnellement sur les deux attributs AGE et REVENU. On note que l'intention du résumé z contient celle de z' et celle de z''.*

$$z'.AGE \subseteq z.AGE \text{ et } z''.AGE \subseteq z.AGE$$

$$z'.REVENU \subseteq z.REVENU \text{ et } z''.REVENU \subseteq z.REVENU$$



– *Relation d'ordre partiel sur l'extension : l'ensemble des n-uplets affectés au résumé z est celui que nous retrouvons dans l'union des n-uplets se trouvant dans z' et dans z''.*

$$R_{z'} \subseteq R_z \text{ et } R_{z''} \subseteq R_z$$

on conclut que $R_z = R_{z'} \cup R_{z''}$.

Les résumés z' et z'' sont généralisés par le résumé z, ce sont deux branches de z dans la hiérarchie. Ils sont plus spécifiques, et le niveau de synthèse qu'ils offrent sur les données est plus réduit que celui que donne le résumé z. D'un autre côté, la représentativité de ces deux résumés z' et z'' est plus faible que celle du résumé z. Cette représentativité par rapport à la base originale est obtenue grâce aux mesures de cardinalités des résumés abordées dans la section 4.2. Ainsi, les cardinalités de l'ensemble de ces trois résumés et suivant la relation qui existe entre eux peuvent être exprimées comme suit, sachant que $card(R_z) = \sum_{ct \in R_z} \omega(ct)$, l'extension R_z contient généralement les n-uplets candidats.

$$|z| > |z'| \text{ et } |z| > |z''|$$

$$card(R_z) > card(R_{z'}) \text{ et } card(R_z) > card(R_{z''})$$

4.3.1.2 Orthogonalité de la hiérarchie

La hiérarchie produite par le système SaintEtiQ est construite d'une façon qu'à chaque niveau, si on considère l'ensemble des fils d'un résumé, la lecture de leurs intentions respectives permette de replacer un n-uplet candidat appartenant au résumé père de manière déterministe dans l'un de ses fils.

Definition 4.5 (Hiérarchie faiblement orthogonale)**.** *Soit z_0 un résumé et $z_1, \ldots z_n$ l'ensemble de ses fils. Alors la hiérarchie $z_0, \ldots z_n$ est dite faiblement orthogonale si :*

$$\forall i, j \in [1, n] \, \exists k \leq m, z_i.A_k \cap z_j.A_k = \emptyset,$$

où m est le nombre d'attributs du résumé z.

Exemple 4.6 (Hiérarchie faiblement orthogonale). *Soit les résumés z_1, z_2, z_3 définis en intention, sur les deux attributs AGE et REVENU par :*

$$
\begin{aligned}
z_1 &= \langle \{0.8/jeune + 0.6/vénérable\}, \{0.4/raisonnable + 0.8/énorme\} \rangle \\
z_2 &= \langle \{0.6/jeune\}, \{0.9/raisonnable + 0.7/indécent\} \rangle \\
z_3 &= \langle \{1.0/enfant + 0.9/vénérable\}, \{1.0/indécent\} \rangle
\end{aligned}
$$

Dans ce cas, z_1 et z_2 ne vérifient pas la propriété énoncée dans la définition 4.5, puisque :

$z_1.AGE \cap z_2.AGE = \{0.6/jeune\}$ *et* $z_1.REVENU \cap z_2.REVENU = \{0.4/raisonnable\}$

Par contre, z_2 et z_3 vérifient cette propriété, car $z_2.AGE \cap z_3.AGE = \emptyset$.

Cette contrainte structurelle forte, est essentielle dans la définition d'un modèle pour chaque coupe sur la hiérarchie dans les meilleures conditions.

4.3.1.3 Hauteur d'un résumé

Definition 4.6 (Hauteur du résumé)**.** *La hauteur d'un résumé notée $haut(z)$ n'est que la profondeur de la plus longue branche du sous-arbre du nœud z. Cette mesure représente la hauteur dans l'arbre du résumé. Par exemple $haut(z) = 0$, z est une feuille.*

Cette mesure est définie afin de pouvoir localiser le niveau auquel appartient un résumé, elle peut effectivement renseigner un utilisateur sur le degré de spécificité d'un résumé suivant la profondeur à laquelle il se situe sur l'ensemble de la hiérarchie.

Les différentes caractéristiques de la hiérarchie sont souvent relatives aux caractéristiques des résumés mêmes, comme leurs cardinalités ou la relation qui existe entre eux. En effet, la relation d'ordre partiel permet aussi d'identifier une structure arborescente sur les connaissances contenues dans les résumés. Quand à la forte contrainte de faible orthogonalité, c'est une forte condition nécessaire au modèle qui sera présenté dans ce chapitre, ainsi qu'à l'algèbre de manipulation des résumés.

4.4 Le modèle multidimensionnel de résumés

La problématique soulevée jusqu'ici dans ce chapitre, concernant le grand volume de la hiérarchie des résumés produite par SaintEtiQ, est celle de comment rendre plus facile le fait d'explorer cette dernière dans un but d'extraction de connaissances ou d'analyse de données.

Notre proposition ici, est celle de la définition d'un modèle basé sur une structure multidimensionnelle pour supporter l'exploration et l'analyse en ligne dans une hiérarchie de résumés linguistiques. Cette structure est un cadre général permettant de donner une définition formelle aux différentes extractions qu'on peut faire sur les résumés de la hiérarchie.

4.4.1 Partition de résumés

Du point de vue de l'utilisateur, chaque coupe de la hiérarchie est une forme différente de la relation d'origine R. Ceci est dû, au fait que chaque coupe contient des résumés différents où chacun représente une partie de la R. Nous voulons construire une vue matérialisée générique à partir de l'ensemble des parties de la relation R. En conséquence, nous proposons de définir la notion de **partition de résumés** sur une collection de résumés qui vérifient quelques propriétés.

Definition 4.7 (Partition de résumés)**.** *Soit $\mathcal{A} = \{A_1, \dots A_m\}$ un ensemble d'attributs. Une partition de résumé P est un ensemble de résumés z construit sur $\mathcal{A}$ et satisfaisant la propriété d'orthogonalité faible.*

La propriété d'orthogonalité faible définie sur la hiérarchie de SaintEtiQ, se traduit sur une partition de résumé P de la manière suivante :

$$\forall (z, z') \in P^2, z \neq z' \quad \exists A \in \mathcal{A}, z.A \cap z'.A = \emptyset \ et, \ ct \in z \cap ct \in z' = \emptyset.$$

Ce qui signifie qu'on ne peut pas avoir deux résumés dans la même partition qui ont la même représentation intentionnelle sur la totalité des attributs. On dit que deux résumés sont conflictuels s'ils ne vérifient pas la propriété d'orthogonalité faible. Il est important de dire qu'une partition de résumés permet un partitionnement réel des données à l'aide des extensions des résumés.

Deux résumés z et z' sont dits conflictuels si, pour chaque attribut, z et z' ont au moins un descripteur en commun, c'est-à-dire si z et z' vérifient :

$$\forall A \in \mathcal{A}, |z.A \cap_F z'.A| > 0$$

Definition 4.8 (Espace de partitions)**.** *Soit $\mathcal{A} = \{A_1, \dots A_k\}$ un ensemble d'attributs. L'espace de partitions noté $\bar{\mathcal{P}}$ est l'ensemble des partitions de résumés construites sur X, $\forall X \subseteq \mathcal{A}$.*

4.4.2 Un niveau d'abstraction

La hiérarchie produite par SaintEtiQ fournit à l'utilisateur un modèle général des données à différents niveaux d'abstraction. Ces niveaux sont appelés *des niveaux de granularité*. Ici nous considérons qu'un niveau d'abstraction ou de granularité correspond à une *coupe* de la hiérarchie.

A partir de cette hiérarchie H_R, il nous est possible de sélectionner un ensemble de résumés pour construire une nouvelle relation $R^\star$ dont le schéma est le même que la relation R. La relation $R^\star$ contient les représentations intentionnelles des résumés contenant les n-uplets réécrits des enregistrements de R, elle doit couvrir la totalité de la relation originale R tel que :

$$R^\star = \sigma(R) \ et \bigcup_{z \in R^\star} = R.$$

Cependant, la taille de $R^\star$ peut être librement choisie. La relation $R^\star$ la plus précise pour représenter R est donnée par l'ensemble de tous les résumés feuilles. A l'opposé, la relation la plus concise est celle qui regroupe un seul résumé, celui représenté par le nœud de la racine. Une relation $R^\star$ de n'importe quelle taille intermédiaire, composée de résumés plus ou moins précis, choisis pour représenter l'intégralité de la relation de base R, peut être calculée à partir de la hiérarchie H_R.

Definition 4.9 (Coupe)**.** *Une coupe C de l'arbre des résumés est un ensemble de résumés vérifiant les deux propriétés suivantes :*

- *l'orthogonalité faible et ,*
- *la complétude $\bigcup_C(R_z) = R$.*

On note $\mathcal{C}(H_R)$ l'ensemble des coupes construites sur l'hiérarchie H_R.

Exemple 4.7 (Une coupe de la hiérarchie). *La figure 4.5, montre un exemple de coupe(s) que nous pouvons extraire à partir de la hiérarchie présentée sur cette figure. La coupe la plus générale est la racine de la hiérarchie z_0 et la plus spécifique est celle qui contient l'ensemble des feuilles $z_* = \{z_i, avec\ haut(z_i) = 0\}$, $haut(z_i)$ est la hauteur d'un nœud résumé dans la hiérarchie (voir définition 4.6).*

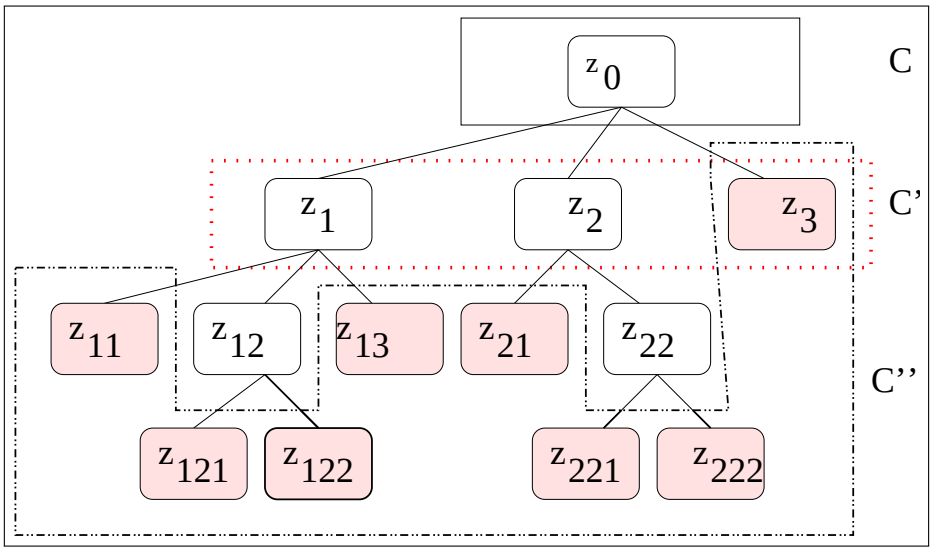


Figure 4.5 – Exemple de coupes sur une hiérarchie

Dans la suite du document, les deux termes *coupe* et *partition de résumés*, seront utilisés pour désigner la même structure. Elle fournit à un niveau d'abstraction donné, un ensemble de résumés muni des deux contraintes de complétude et d'orthogonalité faible.

Génération des coupes

Nous considérons une hiérarchie H_R déjà générée par SaintEtiQ à partir de la base originale R. L'objectif est d'extraire toutes les vues, à différents niveaux

d'abstraction, de cette hiérarchie. Chaque vue est une coupe de la hiérarchie qui couvre la totalité de la base R.

ALG. 4.1 – Calcul des coupes d'une hiérarchie

Entrée: H_R : une hiérarchie de résumés, de racine *root*.
Sortie: un ensemble de coupes.

```
DEBUT
Déclarations : entier i = 0, P_i = ∅,
P̄ ⟵ <> {la liste des coupes}
Appel principal : CalculCoupe( root);
CalculCoupe(nœud)
début CalculCoupe()
si nœud a des voisins alors
   P_i ⟵ nœud + liste des voisins
sinon
   P_i ⟵ nœud
fin si
si nœud a des fils alors
   pour chaque fils du nœud faire
      i ⟵ i + 1
      P_i ⟵ nœudfils
      nœud ⟵ nœudfils
      CalculCoupe(nœud)
   fin pour
sinon
   nœud ⟵ nœudvoisin
   CalculCoupe(nœud)
fin si
P̄ ⟵ P_i
retourner P̄.
fin CalculCoupe()
FIN.
```

La complexité de cet algorithme est de $\mathcal{O}(n)$, n étant le nombre de nœuds de la hiérarchie H_R.

Exemple 4.8 (Les coupes d'une hiérarchie de résumés). *Nous prenons dans cet exemple la hiérarchie présentée dans la figure 4.6, afin d'en extraire toutes les coupes possibles.*

L'ensemble des coupes extraites, noté $\mathcal{C}(H_R)$, comprend la collections de partitions

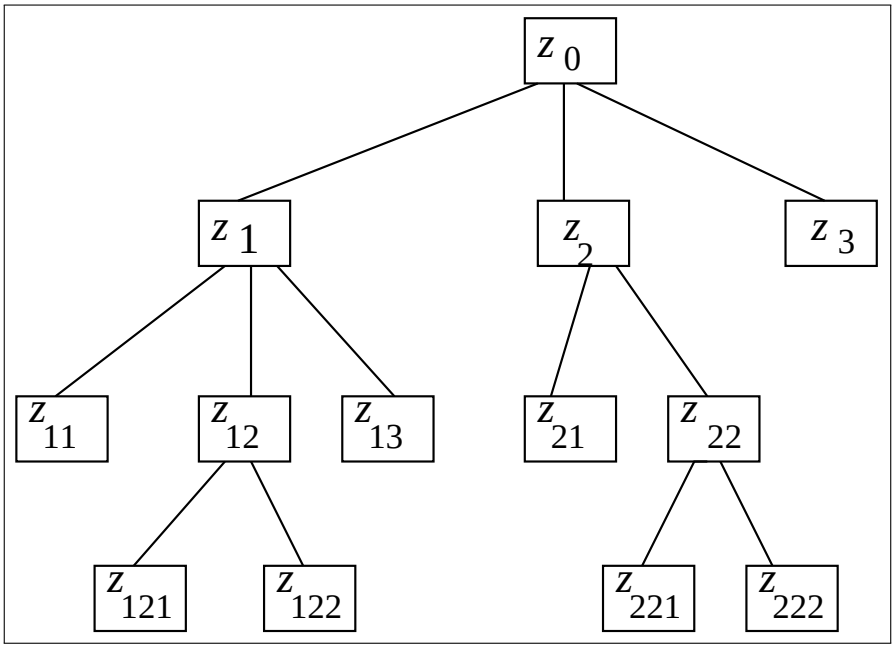


Figure 4.6 – Hiérarchie d'un résumé résultat

suivantes :

$$\mathcal{C}(H_R): \begin{cases} P_0 = & \{z_0\} \\ P_1 = & \{z_1, z_2, z_3\} \\ P_2 = & \{z_{11}, z_{12}, z_{13}, z_2, z_3\} \\ P_3 = & \{z_{11}, z_{121}, z_{122}, z_{13}, z_2, z_3\} \\ P_4 = & \{z_1, z_{21}, z_{22}, z_3\} \\ P_5 = & \{z_1, z_{21}, z_{221}, z_{222}, z_3\} \\ P_6 = & \{z_{11}, z_{12}, z_{13}, z_{21}, z_{22}, z_3\} \\ P_7 = & \{z_{11}, z_{12}, z_{13}, z_{21}, z_{221}, z_{222}, z_3\} \\ P_8 = & \{z_{11}, z_{121}, z_{122}, z_{13}, z_{21}, z_{22}, z_3\} \\ P_9 = & \{z_{11}, z_{121}, z_{122}, z_{13}, z_{21}, z_{221}, z_{222}, z_3\} \end{cases}$$

Cet ensemble de coupes est constitué de dix partitions de résumés qui représentent chacune la base de données R originale avec une précision variable.

L'ensemble des coupes d'une hiérarchie $\mathcal{C}(\mathcal{H}_R)$ est constitué donc des partitions de résumés qui forment une coupe couvrant la relation R. Ces partitions sont représentées dans l'espace de partitions défini sur un ensemble d'attributs $\mathcal{A}$. Nous concluons donc que :

$$\mathcal{C}(\mathcal{H}_R) \subseteq \bar{P}$$

$\bar{P}$ étant l'espace de partition sur R.

La structure de données définie dans ce chapitre, une collection de coupes d'une hiérarchie de résumés, a été inspirée de la structure multidimensionnelle des systèmes décisionnels appelée *cube de données*.

4.4.3 Partition de résumés vs. cube de données

Le modèle proposé dans ce chapitre ne constitue aucunement une extension des cubes de données, mais partage plutôt un objectif commun qui est de fournir une vue synthétique sur des données. Dans la mesure où nous nous sommes inspirés des cubes OLAP pour proposer la structure de partitions de résumés, tout ceci dans le but de supporter un processus d'analyse en ligne, il s'avère important de faire une comparaison entre les deux structures.

Le tableau 4.2 présente donc un comparatif entre les cubes de données et les partitions de résumés sur les critères de base d'une structure multidimensionnelle (fait, mesure, dimension et hiérarchie). Pour mémoire, une comparaison générale des principales similitudes et distinctions entre ces deux structures a été abordée en conclusion de la première partie de ce manuscrit.

Modèle	**Partition de résumés**	**Cube de données**
Fait	n'existe pas / ensemble d'attributs résumés	choix du sujet pour les axes d'analyse
Mesure	degrés d'appartenance - degrés de représentativité description linguistique une mesure qualitative et quantitative	 valeurs agrégées des mesures quantitatives
Dimension	attribut	attribut ou table relationnelle
Hiérarchie	l'espace des attributs multidimensionnels étiquettes linguistiques	existe sur les dimensions intervalles de valeurs

Table 4.2 – Comparaison entre une partition de résumés et un cube de données.

Le modèle tel qu'il vient d'être défini, fournit à l'utilisateur des vues de granularité variable, composées d'un ensemble de résumés. Ces vues ou partitions de résumés sont extraites de la hiérarchie mais il n'existe a priori aucune organisation naturelle sur l'espace de partitions qui nous permette d'évaluer systématiquement la granularité relative d'une partition par rapport à une autre. Nous proposons dans la section suivante, la formalisation d'un ordre total pour classer ces partitions de résumés.

4.5 Organisation des partitions de résumés

Comme nous l'avons déjà abordé dans la section 4.3.1.1, il existe une relation d'ordre partiel sur les résumés de la hiérarchie. Cette relation traduit un phénomène

de généralisation des résumés les uns par rapport aux autres. Il existe donc une relation d'ordre partiel naturelle entre les partitions extraites de la hiérarchie.

Ainsi, l'ensemble des coupes dans $\mathcal{C}(H_R)$ est partiellement ordonné grâce à l'organisation hiérarchique des résumés eux mêmes. Par définition, une partition P_i est plus fine qu'une partition P_j si cette partition P_i détaille des parties de la partition P_j en plus petites classes. Ceci est garantit dans notre cas par la relation d'ordre partiel qui existe entre un résumé et ses résumés fils, les résumés fils étant représentés par des partitions plus fines.

Exemple 4.9 (Relation de généralisation entre les partitions). *Prenons l'exemple de la hiérarchie de la figure 4.6. Dans l'espace de partitions de l'exemple 4.8, nous remarquons que la partition P_0 généralise trivialement toutes les autres partitions. De plus, la partition P_1 généralise la partition P_2, ainsi que la partition P_4. Ces généralisations s'expriment par :*

$$\begin{array}{llll} P_1 \textit{ généralise } P_2 & \textit{parce que } z_1 \in P_1 & ,\{z_{11}, z_{12}, z_{13}\} \subseteq P_2 & \textit{et } z_1 \succ z_{1i}, \forall i \in [1,2,3] \\ P_1 \textit{ généralise } P_4 & \textit{parce que } z_2 \in P_1 & ,\{z_{21}, z_{22}\} \subseteq P_4 & \textit{et } z_2 \succ z_{2j}, \forall j \in [1,2]\,. \end{array}$$

Les deux partitions P_2 et P_4 se généralisent directement en P_1. En revanche, entre P_2 et P_4 il n'existe aucune relation d'ordre. On remarque alors que ces deux partitions sont à un niveau de généralisation comparable.

4.5.1 Relation d'ordre sur les partitions

Notons par l le niveau sur une hiérarchie de résumés, où l_0 correspond au niveau le plus élevé qui contient la racine de l'arbre. Le niveau auquel appartient un résumé est égal au nombre de branches qui forment le chemin vers la racine de l'arbre des résumés. Le niveau de la racine est donc le niveau 0. Ce niveau est retrouvé dans les partitions, on peut dire qu'une partition est d'un niveau l si tous ses résumés sont situé au niveau l de la hiérarchie, par exemple la partition P_1 qui correspond à la coupe C' de la figure 4.5 se situe au niveau 1, dans ce cas on peut dire que la partition P_0 généralise la partition P_1. En revanche, plusieurs partitions contiennent des résumés appartenant à différents niveaux, il est difficile de dire qui de chacune appartient à un niveau supérieur à l'autre. Ceci crée un problème pour établir un ordre entre les partitions, comme il est noté dans l'exemple 4.9, les deux partitions P_2 et P_4 n'ont pas de relation d'ordre qui définit à quel niveau appartient chacune entre elles, ce qui rend difficile la comparaison de l'une par rapport à l'autre.

Nous cherchons donc à établir un ordre total strict sur l'ensemble des partitions d'une hiérarchie de résumés afin de pouvoir comparer leur niveau de généralisation. Cet ordre doit nous permettre de dire que chaque partition de résumés P d'un niveau l_i avec l_i un niveau de la hiérarchie et $0 \leq i \leq n$, est plus générique qu'une partition appartenant à un niveau l_j avec $i \leq j$.

L'ordre total qu'on veut appliquer sur les partitions est en réalité une extension linéaire[5] de l'ordre partiel naturel existant sur les coupes, obtenu grâce à la relation

[5]Dans le cas des ensembles finis ou dénombrables, une extension linéaire « d'un ordre $<$ permet

d'ordre partiel existante sur les résumés. Pour obtenir cette nouvelle relation, l'ensemble des partitions est tout d'abord organisé selon l'ordre partiel naturel en un Graphe Orienté Acyclique (GOA) constitué des différentes coupes de la hiérarchie. La relation d'ordre partiel existant entre deux partitions est notée $\sqsubseteq$:

$$P' \sqsubseteq P \text{ alors } P \text{ généralise } P'$$

Cette relation est formellement définie comme suit :

$$\forall z' \in P', \exists z \in P / z' \preceq z \text{ et } P' \sqsubseteq P, \text{ avec}(P, P') \in \mathcal{C}(\mathcal{H}_{\mathcal{R}})$$

4.5.1.1 Graphe des partitions

La hiérarchie sur laquelle nous travaillons, définie par $H_R = (Z, \preceq)$, Z un ensemble de nœuds résumés et $\preceq$ est la relation de généralisation qui existe entre les nœuds. Nous rappelons cette relation :

$$\forall (z, z') \in Z, z \preceq z' \Leftrightarrow R_z \subset R_{z'}$$

Definition 4.10 (Graphe des partitions)**.** *Le graphe des partitions G est un triplet $G = (\bar{P}, E, \sqsubseteq)$ où :*

- *$\bar{P}$ est un ensemble fini non vide de partitions de résumés.*
- *E est un ensemble fini d'arrêtes.*
- *$\sqsubseteq$ est une fonction d'incidence qui à chaque arête $e \in E$ associe un ordre $P_i \sqsubseteq P_j$ qui signifie que P_j est plus générale que P_i.*

Exemple 4.10 (Graphe des partitions). *Reprenons l'exemple 4.8 de l'ensemble des coupes d'une hiérarchie de résumés. Le graphe orienté acyclique présenté sur la figure 4.7, traduit la relation d'ordre naturelle entre les partitions de la hiérarchie de la figure 4.6.*

4.5.1.2 Tri topologique du graphe des partitions

Le tri topologique du graphe des partitions $G = (\bar{P}, E, \sqsubseteq)$ est une énumération des sommets $p_1, \ldots, p_n$ qui respecte l'ordre de précédence du graphe, c.à.d que si (p_i, p_j) est un arc on doit avoir $i \leq j$.

On note par $\mathcal{O}$ l'ordre total strict appliqué sur le graphe des partitions tel que :

$$\text{si } p_i \mathcal{O} p_j \text{ alors } p_i \text{ est plus générique que } p_j, (i, j) \in [1, \ldots, n]\,.$$

Ce tri topologique a comme objectif de répondre aux besoins d'un utilisateur, qui cherche en analysant les informations contenues dans les résumés à chaque niveau, à connaître la granularité de la représentation propre à lui fournir un bon compromis entre précision et généralisation.

---

d'énumérer (ou de numéroter) les éléments selon l'ordre croissant (pour «), cette énumération étant compatible avec l'ordre d'origine $<$.

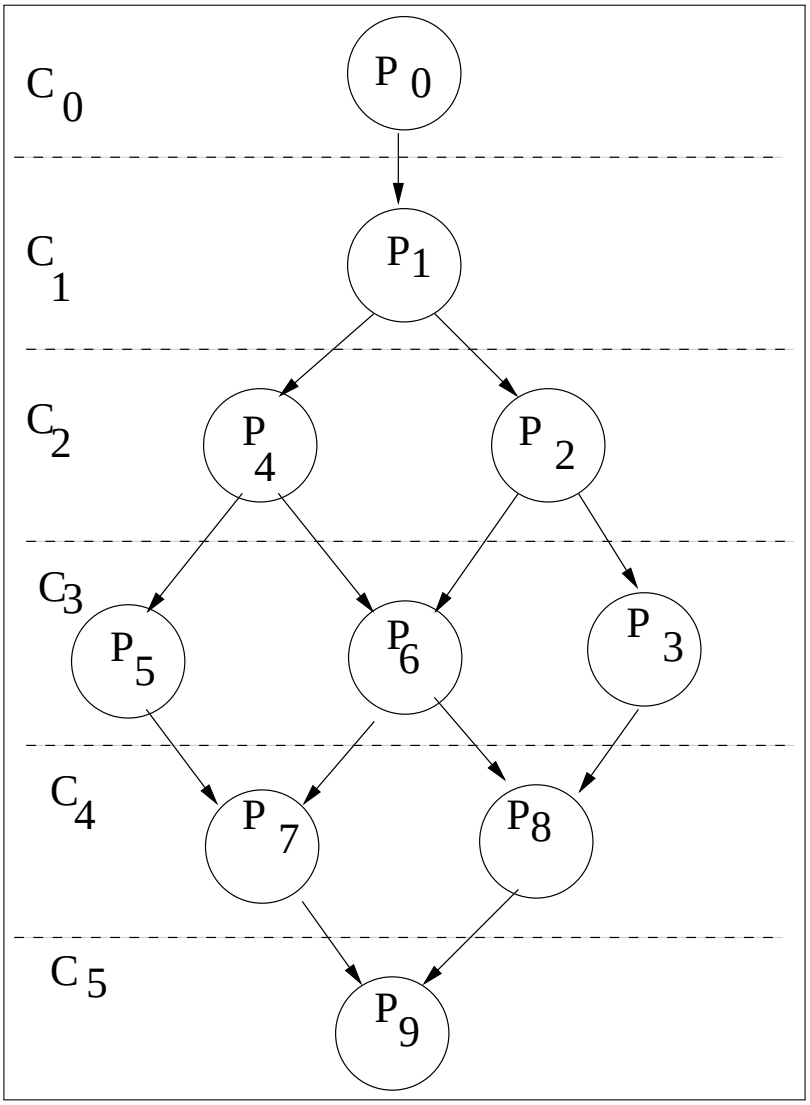


Figure 4.7 – Graphe des partitions

Sur l'exemple 4.10, on remarque que l'organisation des partitions en GOA, ne signifie pas automatiquement l'extension linéaire de la relation d'ordre partiel en un ordre total strict. Ceci est dû à l'existence des *classes de généralisation*, comme le montre le graphe de la figure 4.7.

Classes de généralisation. Une classe de généralisation notée C_i comme sur la figure 4.7, représente un ensemble de partitions qui se trouvent à un niveau de généralisation équivalent.

Afin d'obtenir un ordre total sur l'ensemble des partitions, il est donc nécessaire de trouver un moyen pour ordonner les partitions d'une même classe de généralisation. Or, du point de vue de la topologie du graphe, il est impossible de choisir parmi toutes les extensions linéaires, indépendamment d'une sémantique propre aux partitions de résumés.

4.5.2 Ordre total sur les partitions

Dans les travaux de G. Raschia [83] sur le système SaintEtiQ, plusieurs mesures ont été proposées afin d'évaluer un résumé ou un ensemble de résumés d'une hiérarchie produite par le système. Parmi ces mesures figurent la *spécificité* et la *typicité*.

En ce qui concerne le problème des classes de généralisation et afin d'ordonner deux partitions d'un même niveau de généralisation, les mesures de spécificité et de typicité sont utilisées. Elles offrent un critère de choix entre partitions sur la base d'une sémantique parfaitement définie. En effet, pour savoir si une partition est plus spécifique qu'une autre, il suffit de calculer le degré de spécificité et d'en dériver une typicité globale qui servira a comparer des résumés contenus dans cette partition.

Spécificité. La spécificité notée $Sp(z.A)$ d'un résumé z sur un attribut A indique le degré de précision de $z.A$. Elle est maximale $Sp(z.A) = 1$, dans le cas où $z.A$ est exprimé par un singleton, c'est-à-dire un seul descripteur sur l'attribut A, comme dans les résumés feuilles. Elle est minimale $Sp(z.A) = 0$, quand il s'agit d'un attribut qui est représenté sur tous les descripteurs de D_A^+[6] d'un attribut A. Sur la base de cette mesure concernant les résumés d'une hiérarchie, la mesure de typicité d'une partition entière a été proposée.

4.5.2.1 Typicité d'une partition.

La typicité, notée $\Gamma(P)$, d'une partition de résumés est définie comme une moyenne arithmétique pondérée de valeurs de spécificité des résumés. La typicité est calculée en fonction de la cardinalité relative des résumés, de telle sorte que les résumés faiblement représentés n'offrent qu'une petite contribution à la typicité de la partition, tandis que les résumés représentant une grande part des enregistrements de la relation R ont une forte influence sur le résultat global de la typicité de P.

[6] Domaine réécrit de l'attribut A.

$$\Gamma(P) = \frac{\sum_{z \in P} \mathbf{card}(R_z).\beta(z)}{\sum_{z \in P} \mathbf{card}(R_z)}$$

avec $\beta(z)$ une valeur agrégée de la spécificité des résumés, définie par :

$$\beta(z) = \mathbf{agg}_{A \in \mathcal{A}} \left\{ \mathbf{S_p}(z.A) \right\}, \textit{ où } \mathbf{agg} \textit{ est un opérateur d'agrégation usuel [102].}$$

Ainsi pour l'ordre total sur les partitions, nous proposons de calculer la typicité pour chaque partition au sein d'une même classe de généralisation. La partition ayant une plus faible typicité est celle qui contient les résumés les moins spécifiques. Elle est donc supérieure au sens de l'ordre total strict à établir.

$$\textit{si } \Gamma(P) > \Gamma(P'), \textit{ alors } P' \sqsubseteq P.$$

Exemple 4.11 (Espace de partitions ordonné). *Reprenons l'exemple 4.8, dont le GOA est représenté sur la figure 4.7. Le problème ici se situe au niveau de la comparaison entre les partitions des classes de généralisation suivantes :*
– *C_2, nécessite le calcul de $\Gamma(P_2)$ et $\Gamma(P_4)$.*
– *C_3, nécessite le calcul de $\Gamma(P_3)$, $\Gamma(P_5)$ et $\Gamma(P_6)$.*
– *C_4, nécessite le calcul de $\Gamma(P_7)$ et $\Gamma(P_8)$.*
A priori la partition P_2 contient deux résumés feuilles z_{11} et z_{13} alors que la partition P_4 ne contient que la feuille z_{21}. Il nous permet de conclure que $\Gamma(P_2) > \Gamma(P_4)$ puisque la spécificité des feuilles est maximale et augmente la valeur de typicité d'une partition. Ainsi cela signifie que $P_4 \sqsubseteq P_2$. Sur le même principe nous concluons que $P_6 \sqsubseteq P_5 \sqsubseteq P_3$. En revanche pour les partitions P_7 et P_8, et sans information supplémentaire pour le calcul effectif de la typicité, il paraît difficile de décider laquelle des deux est plus générale elle contiennent le même nombre de résumés qui appartiennent au même niveau de la hiérarchie. Le choix est alors arbitraire ou repose sur une sémantique complémentaire des résumés. Admettons que $P_7 \sqsubseteq P_8$ pour l'exemple, nous aboutissons à l'ordre total suivant de l'ensemble des partitions de résumés :

$$P_0 \sqsubseteq P_1 \sqsubseteq P_4 \sqsubseteq P_2 \sqsubseteq P_6 \sqsubseteq P_5 \sqsubseteq P_3 \sqsubseteq P_7 \sqsubseteq P_8 \sqsubseteq P_9$$

Cette mesure de typicité, si elle permet de traduire la cohésion des résumés, n'est pas suffisante pour donner une réponse toujours satisfaisante pour discriminer les partitions les unes par rapport aux autres. La possibilité d'avoir la même valeur de typicité pour deux partitions existe toujours. Dans ce cas, la fonction qui classe les partitions dans notre système, aura un choix arbitraire entre les deux partitions de résumés. Car, il se trouve qu'elles sont parfaitement similaires au point de vue du niveau de généralisation, de la relation R, qu'elles offrent à l'utilisateur.

4.6 Conclusion

Nous nous sommes intéressés dans ce chapitre à la hiérarchie générée par le processus SaintEtiQ. La problématique soulevée est la suivante : bien que SaintEtiQ soit capable de satisfaire les besoins des décideurs au niveau de la réduction de données relationnelles volumineuses, il n'offre pas un modèle formel qui permet de manipuler les résumés produits organisés sous forme d'une hiérarchie.

L'un des objectifs de ce chapitre, a été de fournir à l'utilisateur une vue, appelée *partition de résumés*, de l'ensemble des données résumées à un niveau d'abstraction donné. Cette vue est représentée par une coupe dans la hiérarchie, elle contient le minimum de résumés avec un maximum d'information. Une partition de résumés est orientée utilisateur, elle lui offre la possibilité d'avoir un ensemble de résumés sur chaque niveau de généralisation de la hiérarchie.

Le modèle défini dans ce chapitre a permis de présenter une structure formelle qui facilitera la manipulation de la hiérarchie des résumés. Ceci est dans le but d'aider l'utilisateur lors d'un processus d'exploration et d'analyse en ligne des résumés. La manipulation des résumés peut répondre à des interrogations comme les suivantes :

- Quels sont les résumés dans ma partition qui sur un attribut précis ont le même degré d'appartenance ?
- Quels sont les résumés qui contiennent les détails d'un résumé donné, se trouvant dans une partition donnée?

Dans le chapitre suivant nous proposons un ensemble d'opérateurs algébriques qui permettent de manipuler l'ensemble des partitions et des résumés linguistiques de la hiérarchie de SaintEtiQ.

Chapitre 5

Une algèbre de manipulation pour les résumés de données

Découvrir c'est bien souvent dévoiler quelque chose qui a toujours été là, mais que l'habitude cachait à nos regards.
— Koestler, Arthur, Le cri d'Archimède.

Au chapitre précédent, un modèle multidimensionnel permettant de réorganiser les résumés dans des partitions, a été proposé. Ces partitions, destinées à un utilisateur qui souhaite analyser et explorer la hiérarchie, contiennent un grand nombre de résumés. Dans ce chapitre, nous présentons un ensemble d'opérateurs algébriques conçus pour manipuler les partitions de résumés extraites de la hiérarchie de SaintEtiQ.

5.1 Introduction à la manipulation des résumés

5.1.1 Objectifs et motivations

Nous avons défini un modèle qui représente une structure multidimensionnelle orientée utilisateur. En effet, le besoin de cet utilisateur est d'être assisté par un outil lui facilitant l'exploration d'une hiérarchie de résumés linguistiques flous. Afin de répondre aux besoins, le modèle proposé doit être équipé d'une algèbre pour manipuler les partitions de résumés. Ainsi la motivation fondamentale qui justifie la définition de ces opérateurs est triple. Elle concerne : le changement de point de vue, l'interactivité et l'analyse.

Pour atteindre cet objectif, nous nous sommes inspirés des opérateurs de manipulation des cubes de données dans les systèmes OLAP. L'ensemble de ces opérateurs a été recensé dans le premier chapitre de ce document. L'étude des systèmes OLAP a montré que l'analyse en ligne est réalisée à l'aide d'un ensemble d'opérateurs algébriques. Cette algèbre fait la force des systèmes OLAP en matière d'exploration et d'analyse des données résumées ou agrégées se trouvant dans les cubes. Par analogie à ces opérateurs de manipulation de données multidimensionnelles, nous proposons dans le cadre de l'exploitation des résumés de bases de données, la définition d'opérateurs capables de modifier le point de vue sur l'information qu'ils explorent.

5.1.2 Proposition

Le travail que nous proposons dans ce chapitre concerne la définition d'une algèbre pour la manipulation des partitions de résumés définies dans le modèle multidimensionnel du chapitre précédent. Plus précisément, ce travail consiste à adapter à cette structure de résumés flous, l'ensemble des opérateurs des trois différentes catégories : les opérateurs classiques [1], les opérateurs de granularité et les opérateurs de restructuration. L'algèbre proposée ici prend en compte les spécificités des résumés de données que nous manipulons et permet d'exploiter les n-uplets qu'ils décrivent. Il s'agit d'un mode d'exploration dédié aux utilisateurs s'intéressant à une vue synthétique et globale de l'information. Ce chapitre présente le noyau de cette algèbre, avec tous les opérateurs d'analyse en ligne adaptés aux partitions de résumés. Il présente aussi une discussion sur les propriétés fondamentales d'une algèbre telles que la complétude, la fermeture et la sémantique. La dernière section du chapitre est consacrée au prototype, orienté utilisateur, développé autour de cette algèbre.

5.1.3 Illustration

L'ensemble des opérateurs algébriques que nous définissons dans ce chapitre est appliqué sur des partitions de résumés. Ainsi, afin d'illustrer chacune des opérations définies, il est souhaitable de disposer d'un exemple complet. Nous utiliserons pour cela, la hiérarchie qui figure dans l'exemple 4.8 du chapitre 3, ainsi que la partition P_3 issue de cette hiérarchie. Nous reproduisons la hiérarchie sur la figure 5.1 et définissons la partition P_3 dans le tableau 5.1. Pour chaque descripteur d est indiqué son degré de satisfaction α_d. La cardinalité $card$ est donnée pour chaque résumé de la partition.

	ACTIVITE	α_d	REVENU	α_d	$card(z)$
z_{11}	*homme d'affaires*	0.7	*misérable*	1.0	0.5
	artiste	0.8	*énorme*	1.0	
z_{121}	*agent de sécurité*	0.9	*misérable*	1.0	0.5
z_{122}	*employé*	0.3	*modeste*	1.0	0.5
z_{13}	*employé*	0.8	*énorme*	1.0	1.0
z_2	*artiste*	0.8	*modeste*	1.0	2.5
			misérable	1.0	
z_3	*homme d'affaires*	0.8	*énorme*	1.0	2.5

Table 5.1 – Présentation de la partition P_3

[1] issus de l'algèbre relationnelle.

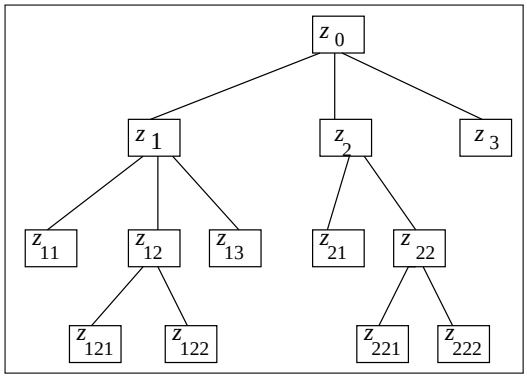


Figure 5.1 – Hiérarchie de résumés

5.2 Le noyau de l'algèbre

5.2.1 Opérations classiques

Nous commençons ici par étudier les opérations dites classiques. Notre but est d'établir une similarité entre ce que nous appelons opérations classiques et les opérations de l'algèbre relationnelle. Ce travail ne consiste pas simplement à reproduire les opérateurs relationnels pour les appliquer aux partitions de résumés, mais plutôt d'adapter ces opérateurs à la nouvelle structure de données multidimensionnelle, et de les munir des capacités d'analyse en ligne propres aux opérateurs de type OLAP. Nous définissons ici, l'ensemble des opérations de sélection, projection, fusion, produit cartésien, jointure ainsi que les opérateurs ensemblistes.

5.2.1.1 Opération de sélection

Dans l'algèbre relationnelle, l'opérateur de sélection s'applique sur un ensemble d'attributs avec un prédicat logique que doivent satisfaire les n-uplets sélectionnés. Dans cette section, nous essayons d'étendre cette opération à notre modèle de données. Dans notre cas nous pouvons employer des prédicats flous, étant donné que, nous travaillons sur des résumés de données qui utilisent des concepts issus de la théorie des sous-ensembles flous.

L'opérateur de sélection permet d'extraire un ensemble de résumés à partir d'une partition de résumés sans aucune mise à jour de la description intentionnelle ou extensionnelle des résumés.

Definition 5.1 (Sélection)**.** *L'opérateur de sélection noté σ appliqué sur une partition de résumés P à partir de la relation R, donne une nouvelle partition de résumés P' telle que :*

$$Sélection\,\langle partition\ \ de\ \ résumés\,\rangle\,\langle prédicat\rangle$$

$$P' = \sigma(P, pred(z)),$$

où $pred(z)$ est le prédicat de sélection qu'on peut appliquer sur :

– *les propriétés des résumés (Dice), ou*
– *les valeurs d'attributs (Slice).*

Dice.

Le premier type de filtrage correspond à une sélection sur les propriétés des résumés. En se basant sur l'organisation hiérarchique des résumés, nous pouvons exploiter plusieurs propriétés, comme par exemple, une contrainte de granularité exprimée par la mesure de hauteur d'un résumé (voir définition 4.6), ou encore un prédicat de sélection sur le degré de représentativité d'un résumé, exprimé par sa cardinalité ($card_z$). Dans le cas où on utilise la hauteur, le prédicat s'écrit : $h(z)$ *op* $n, n \in \mathbb{N}$ avec $op \in \{=, <, >, \leq, \geq\}$.

Ainsi, en appliquant un prédicat de sélection sur les résumés, l'opération est appelée *dice*. Elle permet de spécifier une requête selon des critères caractérisant le résumé, comme sa représentativité ou sa granularité.

– *La représentativité d'un résumé.* La cardinalité relative d'un résumé $card(z)$ représente le poids du résumé par rapport à la base de données originale. Cette mesure peut définir un critère de sélection qui exprime la représentativité d'un résumé. Ce critère faisant l'objet d'un prédicat de sélection permettra d'identifier la liste des résumés dans une partition qui se rapprochent le plus, par leur représentativité, de la base originale. L'exemple 5.12, montre une opération du *dice* sur un prédicat de représentativité.

Exemple 5.12 (Dice sur la représentativité). *On considère la partition P_3 présenté sur le tableau 5.1, et on formule le prédicat suivant :*

$$pred(z) : card(z) = \max_{z_i \in P}(card(z_i))$$

Le prédicat ainsi défini sélectionne dans une partition les résumés dont la représentativité est maximale, ce qui veut dire, dont les cardinalités sont les plus élevées. Le résultat de cette opération sélectionnera la partition suivante : $P' = \{z_2, z_3\}$.

– *La granularité d'un résumé.* On considère le degré du résumé comme mesure approchée de la granularité des résumés constituant une partition à un niveau d'abstraction. Ce critère nous permet de distinguer les résumés dont la granularité vérifie la contrainte, et notamment d'identifier un ensemble de résumés homogènes sur une partition donnée.

Exemple 5.13 (Dice sur la granularité). *Prenons la hiérarchie de résumés, se trouvant sur la figure 5.5, si nous voulons sur une partition donnée $P = \{z_{11}, z_{12}, z_{13}, z_{21}, z_{22}, z_3\}$ rechercher seulement les feuilles de cette partition, $pred(z)$est défini comme suit $h(z) = 0$, le dice s'ecrit $P' = \sigma(P, h(z) = 0)$, ce qui donnera en résultat la partition $P' = \{z_{11}, z_{13}, z_{21}, z_3\}$.*

Slice.

Le second type de filtrage est celui de la sélection sur les valeurs floues des attributs décrivant les résumés ainsi que sur leurs degrés d'appartenance. Cette opération est définie par l'expression suivante :

$$z.A\ \theta\ \tilde{v},$$

où θ est un opérateur d'ensembles flous comme $=_F, \subseteq_F, \cap_F, \subset_F, etc$, et $\tilde{v} \in \mathcal{F}(D_A^+)$, est un sous-ensemble flou. On peut généraliser cette même expression à la suivante :

$$f(z.A)\ \theta\ \tilde{v},$$

où f peut être un ensemble de caractéristiques floues, comme ($haut(z.A)$ ou $card(z.A)$), pour plus de détails concernant les sous-ensembles flous le lecteur est invité à consulter l'ouvrage de Zadeh [105]. Une seconde forme de filtrage sur les valeurs d'attributs s'exprime de la façon suivante :

$$z.A\ \theta\ z.B,$$

ici on définit un prédicat de sélection qui compare deux descriptions sur deux attributs différents A et B au sein du même résumé z.

Exemple 5.14 (Slice). *Soit l'expression suivante :*

$$P' = \sigma(P, z.REVENU \subseteq_F \{1.0/norme, 1.0/modeste\})$$

En appliquant cette opération de slice sur la partition de résumés P_3 du tableau 5.1, le résultat est le suivant :

$$P' = \{z_{11}, z_{122}, z_{13}, z_2, z_3\}.$$

Sur cette même partition, P_3 du tableau 5.1, nous pouvons aussi chercher à extraire les résumés qui contiennent l'étiquette linguistique misérable *dans leurs descriptions intentionnelles. Cette requête sera exprimée par la sélection suivante :*

$$P' = \sigma(P, z.A \cap_F \{misérable\} \neq 0),$$

le résultat dans ce cas est : $P' = \{z_{11}, z_{121}, z_2\}$.

Combinaison des prédicats de sélection

Afin d'exprimer des prédicats plus complexes dans l'opération de sélection, nous utilisons des combinaisons conjonctives ($pred_1(z) \wedge \cdots \wedge pred_k(z)$) ou disjonctives ($pred_1(z) \vee \cdots \vee pred_k(z)$)de prédicats de sélection.

Exemple 5.15 (Expression d'une requête de sélection). *Les différents cas de l'opération de sélection que nous venons d'étudier nous permettent d'écrire*

une expression algébrique qui peut fournir une réponse à une interrogation telle que " trouver sur la partition de résumés P_3, (voir le tableau 5.1), les résumés les plus spécifiques en terme de granularité, et qui représentent les gens de faible revenu". Pour cette requête, nous proposons l'expression suivante :

$$\begin{aligned} P' &= \sigma(P_3, (z.REVENU \cap_F \{misérable\} \neq 0) \wedge (h(z) = 0)) \\ &= \{z_{11}, z_{121}\}. \end{aligned}$$

Ainsi, la réponse à cette requête correspond aux n-uplets des résumés feuilles z_{11} et z_{121}, pour la partition P_3.

Notons que l'opération de sélection ne modifie en aucun cas la description intentionnelle des résumés manipulés, ce qui signifie que ni les extensions des résumés ni leurs cardinalités ne seront affectées par l'application d'un des opérateurs de sélection.

5.2.1.2 Opération de fusion

L'opération de fusion consiste à regrouper les descriptions intentionnelles d'un ensemble de résumés d'une partition en agrégeant, pour chaque attribut, les degrés d'appartenance des descripteurs similaires, et par l'union des ensembles flous.

Definition 5.2 (Fusion)**.** *Soit $P = \{z_1, .., z_k, .., z_n\}$, une partition de résumés. On définit l'opération de fusion comme suit :*

$$\begin{aligned} P' &= merge(P, \{z_1, .., z_k\}), \\ &= \{z^\star, z_{k+1}, ..., z_n\} \end{aligned}$$

Où le résumé $z^\star = z_1 \cup_F z_2 \cup_F \ldots \cup_F z_k$, remplace l'ensemble des résumés à fusionner $\{z_1, .., z_k\}$. L'union floue $\cup_F$ est appliquée sur chaque attribut en considérant les sous-ensembles flous $z_i.A$.

Par exemple, la fusion $merge(P, \{z_i, z_j\})$ avec $z_i = \langle \{\alpha_a/d_1\}, \{\alpha_b/d_2\} \rangle$ et $z_j = \langle \{\alpha_c/d_1\}, \{\alpha_d/d_3\} \rangle$, donne naissance au résumé $z^\star = \langle \{max\{\alpha_a, \alpha_c\}/d_1\}, \{\alpha_b/d_2\}, \{\alpha_d/d_3\} \rangle$.

Nous rappelons que toute partition de résumés doit respecter la contrainte forte sur la faible orthogonalité. Pour ceci, le nouveau résumé qui représente la fusion ne doit, en aucun cas, être conflictuel avec un autre résumé de la nouvelle partition P'. Nous supposons alors que l'opération de fusion ne peut être appliquée que si le résumé $z^\star$ est non conflictuel avec un autre résumé, ceci nécessite que $\forall z \in P', \ z \neq z^\star$.

Comme indiqué dans la définition 5.2, l'opération de fusion manipule l'intention des résumés. Ainsi, un résumé résultant d'une fusion contient les n-uplets de l'ensemble des résumés fusionnés. L'extension du résumé résultat est alors égale à l'union des extensions des résumés fusionnés.

$$R_{z^\star} = R_{z_1} \cup \cdots \cup R_{z_k}$$

Par ailleurs, la cardinalité de chaque résumé fusionné n'est pas affectée non plus. La cardinalité du résumé $card(z^\star)$ est la suivante :

$$card(z^\star) = card(z_1) + \cdots + card(z_k)$$

Exemple 5.16 (Fusion). *Afin d'illustrer cette opération de fusion, nous reprenons la partition P_3 donnée par le tableau 5.1. Supposons que dans un scénario d'interrogation, l'utilisateur a déjà fait une sélection sur les résumés feuilles de la partition avec une description intentionnelle contenant le descripteur* employé *sur l'attribut ACTIVITE. Et considérons, le résultat de cette sélection sur la partition manipulée, soit les deux résumés z_{13} et z_{122}. Maintenant l'utilisateur souhaite fusionner ces résumés afin de pouvoir avoir une vue de synthèse sur la nouvelle partition constituée de ces deux résumés. Le tableau 5.2 montre comment on peut fusionner ces deux résumés. Le résumé $z^\star$ représente les n-uplets des deux résumés feuilles, sur l'attribut ACTIVITE, du descripteur employé, soit $\alpha(employé) = 0.8$.*

	ACTIVITE	α_d	REVENU	α_d	$card(z)$
z_{13}	*employé*	0.8	*énorme*	1.0	1.0
z_{122}	*employé*	0.3	*modeste*	1.0	0.5
merge(P_3, $\{z_{13}, z_{122}\}$)					
$\mathbf{z}^\star$	*employé*	0.8	*misérable* modest	1.0 1.0	1.5

Table 5.2 – Fusion sur la partition de résumés P_3

5.2.1.3 Opération de projection

Rappelons qu'en algèbre relationnelle, l'opération de projection π consiste à sélectionner un sous ensemble d'attributs $A_1, A_2, \ldots, A_n$ de la relation R et se note $\pi_{A_1,A_2,\ldots,A_n}(R)$. Pour l'adaptation de cette opération aux partitions de résumés, nous appelons projection toute opération qui consiste à réduire le nombre d'attributs qui décrivent les résumés d'une partition. Le but étant d'éliminer un ensemble d'attributs dans une partition de résumés.

Definition 5.3 (Projection)**.** *Soit P, une partition de résumés. On définit l'opérateur de projection comme suit :*

$$\begin{aligned} P' &= \Pi_{A_1,..,A_k}(P) \\ P' &= \{z'/\exists z \in P \wedge z' = \pi_{A_1,..,A_k}(z)\} \end{aligned}$$

où la projection sur les résumés est $\pi_x(\langle x, y\rangle) = \langle x\rangle$.

L'impact de cette opération sur l'extension des résumés est nul, ceci est dû au fait que les n-uplets restent les mêmes bien que le nombre d'attribut diminue. Ainsi, la description extensionnelle et les cardinalités des résumés restent inchangés après application d'une projection sur un ensemble de résumés.

L'opération de projection, telle qu'elle est présentée à la définition 5.3, est relativement évidente. Néanmoins, la création de résumés conflictuels n'est pas écartée. Ce problème peut survenir comme pour l'opération de fusion, si la partition résultante ne vérifie pas la propriété de faible orthogonalité. L'exemple suivant illustre le cas de résumés conflictuels. La représentation du résumé dans ce tableau n'est volontairement pas complète puisqu'on ne s'intéresse ni aux degrés de satisfaction ni aux cardinalités des résumés.

Exemple 5.17 (Projection conflictuelle). *L'application de l'opération de projection sur les attributs* (B, C) *nous renvoie les mêmes descripteurs* b_1 *et* c_1 *sur les deux résumés supposés différents* z_1 *et* z_2*. Le résultat de la projection ne peut être une partition de résumés parce qu'il ne vérifie pas la propriété d'orthogonalité faible. En effet,* $\pi_{B,C}(z_1, z_2)$*,* $\pi_B(z_1, z_2)$ *et* $\pi_C(z_1, z_2)$ *donnent des résumés conflictuels, i.e. avec des descriptions similaires.*

	A	B	C
z_1	a_1 a_2	b_1	c_1
z_2	a_3	b_1	c_1

Table 5.3 – Résumés représentés sur les attributs A, B et C

Afin d'éviter un tel problème, nous avons simplement mis une contrainte qui interdit les projections conflictuelles. Seuls les projections ayant en résultat une collection de résumés respectant les deux propriétés d'une partition de résumés sont possibles.

5.2.1.4 Adaptation des opérations relationnelles binaires

Jusqu'ici, nous nous sommes intéressés à traduire dans l'algèbre proposée, destinée aux résumés, des opérateurs relationnels qui agissent sur une seule partition. Nous les qualifions d'opérateurs unaires. D'autres opérateurs de type binaire sont définis, ils vont générer une partition à partir d'une paire de partitions existantes. Ces opérateurs sont : le produit cartésien, la jointure et les opérations ensemblistes.

Opération du produit cartésien

Le produit cartésien est un opérateur binaire qui consiste à combiner les résumés d'une partition de résumés avec tous les résumés d'une autre partition.

Cette combinaison est réalisée en créant des résumés sur l'ensemble des attributs des deux partitions.

Definition 5.4 (Produit Cartésien)**.** *Soient P et P', deux partitions de résumés.On note $P'' = P \times P'$ le produit cartésien des partitions P et P' défini par :*

$$P'' = \left\{ \langle z, z' \rangle \mid z \in P \ \wedge \ z' \in P' \right\}$$

Le couple $\langle z, z' \rangle$ est donc représenté sur les attributs de z et de z', ce qui va certainement changer la description intentionnelle du nouveau résumé résultat du produit du couple $\langle z, z' \rangle$. Ainsi, l'extension et la cardinalité correspondantes à l'opération du produit cartésien seront influencées par les n-uplets des deux résumés considérés, elles sont exprimées comme suit :

$$\begin{array}{lll} R_{z''} & = R_z & \otimes \ R_{z'} \\ card(z") & = card(z) & \times \ card(z') \end{array}$$

Opération de jointure

L'opération de jointure est employée pour connecter des résumés de deux partitions. Le résultat de cet opérateur de jointure est une nouvelle partition de résumés qui vérifie la propriété de faible orthogonalité. On peut noter que la jointure est un cas particulier du produit cartésien qui utilise un prédicat de jointure pour filtrer les résumés du résultat.

Definition 5.5 (Jointure)**.** *L'opérateur join consiste à agréger deux partitions de résumés P et P' selon leurs descriptions intentionnelles suivant un prédicat de jointure mettant en jeu un attribut pivot dans chaque partition. Elle est définie comme suit :*

$$P^* = P \underset{pred(z.A, z'.A')}{\bowtie} P'$$

où $pred(z.A, z'.A') = (z.A)\ \theta\ (z'.A')$, θ est un opérateur de sous-ensembles flous, tel que $(=_F, \subseteq_F, \subset_F, etc)$, $z \in P$ et $z' \in P'$.

Soient les deux partitions de résumés suivantes : $P = \{z_1, .., z_n\}$ et $P' = \{z'_1, .., z'_{n'}\}$. Le résultat de l'opération de jointure sur les attributs $(z.A, z'.A')$ est une nouvelle partition de résumés notée $P^\star = \{z^\star | z^\star.A\ \theta\ z^\star.A'\}$, où $z^\star = \langle z.A, z.B, ..., z'.A', z'.B', ... \rangle$ avec $z = \langle z.A, z.B, \dots \rangle \in P$, $z' = \langle z'.A', z'.B', \dots \rangle \in P'$.

Dans cette opération de jointure on travaille avec l'intention des résumés de chacune des partitions à joindre. L'impact de cette opération sur l'extension et la cardinalité des résumés dans $P^\star$ est le suivant :

$$\begin{array}{ll} R_{z^\star} & = R_z \otimes R_{z'} \\ card(z^\star) & = card(z) \times card(z') \end{array}$$

Exemple 5.18 (Opération de jointure de deux partitions). *Soient les deux partitions de résumés P_1 et P_2 telle que :*

- $P_1 = \{z_1, z_2\}$, *définie sur l'attribut REVENU avec les descripteurs (bas, moyen et élevé) et sur l'attribut ACTIVITE avec les descripteurs (artiste, homme d'affaires et étudiant).*
- $P_2 = \{z'_1\}$, *définie sur l'attribut SALAIRE avec (bas, moyen et élevé) et sur l'attribut AGE avec (jeune, âgé).*

Les descriptions intentionnelles des résumés sont :

- $z_1 = \langle 0.8/bas, 0.5/étudiant\rangle$
- $z_2 = \langle 0.7/élevé, 1.00/artiste\rangle$
- $z'_1 = \langle 1.00/élevé, 0.8/\ jeune\ \rangle$

On écrit la jointure :

$$P^* = P_1 \underset{z_1.REVENU \subseteq_F z'_1.SALAIRE}{\bowtie} P_2$$

$z.A \subseteq_F z'.A'$, *signifie que,* $\alpha_d(z.A) \leq \alpha_d(z'.A')$.

$$\begin{aligned} P^\star \quad = \quad & \{\langle 1.00/élevé, 0.5/étudiant, 0.8/jeune\rangle\ , \\ & \langle 1.00/élevé, 1.00/artiste, 0.8/jeune\rangle\} \end{aligned}$$

Opérations ensemblistes

Nous allons adapter dans ce qui suit les opérateurs ensemblistes à notre modèle de partitions de résumés. Nous définissons les opérateurs d'union, d'intersection et de différence. Pour ceci nous avons besoin de préciser la notion d'*union-compatibilité.*

Definition 5.6 (Union-compatibilité)**.** *On dit que deux partitions de résumés définies sur le même ensemble d'attributs sont union-compatible si tous les résumés de la première partition sont non conflictuels avec tous les résumés de la deuxième partition.*

- *Union.* On considère deux partitions de résumés P et P' qui sont union-compatibles. On note par $P'' = P \cup P'$ l'union des partitions et on la définit : $P'' = \{z | z \in P \vee\ z \in P'\}$.

- *Intersection.* L'intersection de deux partitions de résumés P et P' est notée $P'' = P \cap P'$ et elle est définie comme suit : $P'' = \{z|\ z \in P\ \wedge z \in P'\}$. L'intersection de deux partitions peut être exprimée comme un cas spécial d'une opération d'equi-jointure totale. Une équi-ointure totale est une opération de jointure dont le prédicat est fondé sur l'opérateur d'égalité.

– *Différence.* La différence de deux partitions de résumés P et P' qui sont union-compatible produit une partition de résumés qui se trouvent dans P et non dans P'. On la définit par : $P'' = \{z | z \in P \wedge z \notin P'\}$.

Pour les trois opérations ensemblistes, ni l'extension des résumés ni leurs cardinalités ne sont concernés par des mises à jour après l'application d'une de ces opérations. Cela est conforme au fait que les résumés sont manipulés dans leur intégralité, sans modification sur leurs intentions ou sur les n-uplets qui les constituent.

5.2.2 Opérations de granularité

Nous allons nous intéresser dans cette seconde partie, à la deuxième catégorie d'opérateur de l'algèbre pour l'analyse en ligne de résumés. Cette catégorie concerne les opérations qui agissent sur la granularité des données en utilisant leur organisation hiérarchique.

Comme on l'a déjà évoqué, l'ensemble des partitions extraites d'une hiérarchie de SaintEtiQ est ordonné. Ceci est dû à la relation d'ordre partiel qui existe entre les résumés de la hiérarchie. Grâce à cette relation d'ordre, nous pouvons définir des opérateurs de généralisation et de spécialisation afin de faciliter à l'utilisateur la navigation dans les partitions en explorant les différents niveaux de granularité de la hiérarchie. Ces opérateurs sont appelés *opérateurs de granularité*, et ils sont au nombre de deux.

Nous définissons dans ce qui suit, les deux opérateurs *roll-up* et *drill-down*, que nous considérons comme une adaptation, à notre modèle de partition de résumés, des opérateurs multidimensionnels de type OLAP qui agissent sur la granularité des cubes de données.

Nous rappelons, que les partitions sont pré-calculées et ordonnées avant de pouvoir les manipuler. Et les opérateurs de granularité sont définis sur la hiérarchie initiale générée par SaintEtiQ, cette hiérarchie n'ayant subi aucune modification par l'application d'autres manipulations.

5.2.2.1 Opération de généralisation

Nous définissons l'opérateur *Roll-up* dans l'objectif de visualiser des données agrégées d'un ou de plusieurs résumés vers des résumés plus génériques. Cet opérateur prend en entrée une partition et un ensemble de résumés à généraliser. Il fournit en sortie une partition contenant les résumés qui généralisent ceux passés en paramètre.

Definition 5.7 (Roll-up, ou forage vers le haut)**.** *L'opérateur roll-up est une généralisation d'une partition de résumés à un niveau de granularité plus élevé. Il est définit comme suit :*

$$\begin{aligned} P' &= Roll-up(P = \{z_1, \ldots, z_i, z_{i+1}, \ldots, z_n\}, \{z_1, \ldots, z_i\}) \\ P' &= \{z', z_{i+1}, \ldots, z_n\} \end{aligned}$$

tel que $\forall k \in \{1, \ldots, i\}$, $z_k \preceq z'$, *(*z' *généralise directement* z_k*).*

Suivant la relation d'ordre qui existe entre les partitions, nous constatons qu'une partition peut être généralisée de différente manière, étant donné qu'un résumé est généralisable par tous ses antécédants jusqu'à la racine. Mais dans l'application des opérateurs de granularité, nous nous déplaçons toujours vers la première partition qui peut répondre à notre opération de généralisation. Ce qui revient à chercher le parent direct de l'ensemble des résumés à généraliser. Pour une meilleure compréhension de l'application de cet opérateur de généralisation sur un ensemble de partitions, nous proposons l'exemple suivant.

Exemple 5.19 (L'opérateur *roll-up*).
Nous reprenons toujours l'exemple donné dans la figure 5.1, ainsi que l'ensemble des coupes $\mathcal{C}(\mathcal{H}_R)$ *qui le représente. On considère que l'utilisateur est en train de manipuler la partition* P_7 *qui contient un certain nombre de résumés, par exemple* (z_{221}, z_{222}) *dont il souhaite connaître les parents afin d'avoir une vision plus générale. L'opérateur* roll-up *lui permet d'avoir la partition qui généralise cette partition* P_7 *sur les résumés* (z_{221}, z_{222})*. D'après l'ensemble des partitions, extrait de l'espace de la hiérarchie considérée, ce sera* P_6*.*

$$P_6 = Roll - up(P_7, \{z_{221}, z_{222}\}),$$

avec :

$$z_{221}, z_{222} \preceq z_{22} \subseteq P_6 \quad et\ z_{221}, z_{222} \subseteq P_7 \tag{5.1}$$

$$alors, P_6 \sqsubseteq P_7 \Rightarrow P_6\ généralise\ P_7.$$

$$\mathcal{C}(H_R^\star): \begin{cases} P_4 = & \{z_1, z_{21}, z_{22}, z_3\} \\ P_5 = & \{z_1, z_{21}, z_{221}, z_{222}, z_3\} \\ P_6 = & \{z_{11}, z_{12}, z_{13}, z_{21}, \underline{z_{22}}, z_3\} \\ P_7 = & \{z_{11}, z_{12}, z_{13}, z_{21}, \underline{z_{221}, z_{222}}, z_3\} \end{cases}$$

De la même façon on peut exprimer d'autres généralisations sur l'ensemble de l'espace de partitions de résumés présenté dans 5.1 par :

$$P_2 \quad = \quad Roll - up(P_3, \{z_{121}, z_{122}\})$$

5.2.2.2 Opération de spécialisation

La spécialisation est l'opération inverse de la généralisation. Cette opération sera réalisée à l'aide de l'opérateur *drill-down*, appelé en français opérateur de forage vers le bas. Ce dernier consiste à explorer une hiérarchie de résumés, en partant des résumés ou des partitions les plus génériques pour aller chercher leurs détails se trouvant dans des partitions plus spécifiques. Ainsi, l'utilisateur peut

explorer la hiérarchie des résumés en ayant la possibilité, grâce à l'opérateur *drill-down*, de raffiner progressivement ses manipulations ou ses requêtes, en allant dans les résumés les plus fins.

Definition 5.8 (Drill-down, forage vers le bas)**.** *L'opérateur drill-down consiste en la spécialisation d'une partition de résumés par raffinage d'un résumé donné. Nous le définissons pour $z \in P$ comme suit :*

$$P' = Drill - down(P = \{z_1, z_2 \ldots, z_n\}, z_1)$$
$$P' = \{z'_1, \ldots, z'_n, z_2, \ldots, z_n\}$$
$$\forall i \in [1, m], z'_i \preceq z_1, z_1 \textit{ est directement spécialisé par } z'_i.$$

Exemple 5.20 (L'opérateur *drill-down*). *Inversement à l'exemple 5.19. Nous cherchons la partition qui contient les résumés fils de z_{22} à partir de la partition P_6. Nous retrouvons, suite à l'application d'une opération de forage ves le bas, la partition P_7. L'application de l'opérateur* drill-down*, est donc formulée comme suit :*

$$P_7 = Drill - down(P_6, z_{22}),$$

en prenant en considération, la relation d'ordre entre les résumés de ces deux partitions, tel que mentionné dans l'expression 5.1 de l'exemple 5.19.

Toutes les partitions ayant la relations $P_i \sqsubseteq P_j$ sont des candidates valables pour une opération de généralisation/spécialisation, sauf que dans les opérateurs de granularité définis dans notre algèbre nous choisissons d'aller dans les successeur/antécédent immédiat ou direct du résumé ou des résumés passés en paramètre d'une opération de roll-up ou de drill-down.

5.2.3 Opérations de restructuration

Nous allons nous intéresser maintenant à la dernière catégorie d'opérations d'analyse en ligne, soit les opérateurs de restructuration. Nous adapterons ici seulement les opérateurs qui ont un intérêt vis-à-vis de la manipulation des partitions et des résumés. Pour ceci, nous avons délibérement ignoré la traduction de certains opérateurs qui, à notre avis, n'apportent aucun avantage à l'utilisateur lors de l'exploration de la hiérarchie ou de la manipulation d'une partition. Nous nous sommes concentré sur la définition de quelques opérateurs qui vont agir sur la représentation multidimensionnelle d'une partition. Nous expliquons le choix des opérateurs définis, et nous illustrons chaque définition par un exemple.

5.2.3.1 Représentation multidimensionnelle d'une partition de résumés

Avant de proposer les opérations agissant sur la structure d'une partition de résumés, pour améliorer la visualisation et la navigation dans le résumé, nous commençons d'abord par définir un schéma général pour la visualisation multidimensionnelle d'une partition de résumés.

Pour ceci, nous choisissons une représentation tabulaire. Il s'agit d'un mode de visualisation simple et intuitif auquel les utilisateurs sont habitués dans un contexte décisionnel. C'est aussi la représentation des données traditionnellement adoptée comme le soulignent les auteurs de [2, 95].

Le tableau 5.4 montre la représentation tabulaire d'une partition de résumés, cette représentation est organisée de la manière suivante :

- On affiche en premier le nom de la partition analysée (sujet analysé), soit dans le tableau 5.4, la partition P.
- La première colonne contient les identifiants des résumés de la partition.
- Les colonnes représentent les descripteurs linguistiques $\{d_1 \ldots, d_k\}$ et les attributs $\{A_1, \ldots, A_n\}$.
- Les mesures se trouvent à l'intersection des lignes/colonnes et contiennent des valeurs floues comme le degré d'appartenance α ou la cardinalité *card*. Ces valeurs sont représentées quand elles existent c-à-d quand un attribut est bien décrit par le descripteur concerné (en colone) sur le résumé en question (en ligne). Ce qui explique l'existence de cellules vides (voir la figure 5.5).

Notons que les colonnes représentées en premier sont celles qui peuvent être visualisées à l'écran, les autres s'affichent quand l'utilisateur désire de les défiler.

P	A_1			A_2			...	A_n
	d_{1A_1}	d_{2A_1}	d_{3A_1}	d_{1A_2}	d_{2A_2}	d_{3A_2}		
z_1	$\langle \alpha_d, card \rangle$							
z_2	$\langle \alpha_d, card \rangle$							
z_3	$\langle \alpha_d, card \rangle$							
...	$\langle \alpha_d, card \rangle$							

Table 5.4 – Présentation tabulaire d'une partition de résumés

Afin de montrer le résultat d'une telle représentation sur une partition, nous proposons l'exemple 5.21 qui détaille la partition P_3 déjà présentée dans le tableau 5.1. Les cellules vides sur le tableau 5.5 représentent des valeurs inexistantes sur un descripteur (colonne) pour le résumé correspondant (ligne), mais seulement une manière plus simple de proposer un tableau de représentation d'un niveau général.

Exemple 5.21 (Présentation tabulaire d'une partition de résumés). *La présentation multidimensionnelle de la partition P_3 (voir 5.1) consiste à mettre au premier plan les attributs (dimensions) que l'utilisateur souhaite analyser. La table 5.5 offre cette présentation. Nous y découvrions seulement les deux attributs en cours d'analyse (ACTIVITE et REVENU), ainsi que les descripteurs correspondants. On a choisi par simplification de présenter sur ce tableau seulement le degré d'appartenance α comme mesure.*

Sur la base de cette présentation multidimensionnelle d'une partition, nous

SUM	ACTIVITE				REVENU		
	artiste	*agent.S.*	*employé*	*homme d'affaires*	*misérable*	*modeste*	*énorme*
z_{11}	0.8			0.7	1.0		1.0
z_{121}		0.9			1.0		
z_{122}			0.3			1.0	
z_{13}			0.8				1.0
z_2	0.8					1.0	1.0
z_3				0.8			1.0

Table 5.5 – Présentation multidimensionnelle de la partition de résumés P_3

allons définir un ensemble d'opérateurs de restructuration comme la rotation ou le tri.

5.2.3.2 Opération de rotation

La rotation exprimée à l'aide de l'opérateur *rotate (pivot)*, est utilisée habituellement dans l'algèbre multidimensionnelle pour analyser les données dans diverses perspectives. Cet opérateur s'applique aux dimensions d'un cube de données. Par analogie, une dimension correspond à un attribut d'une partition de résumés. La rotation sur les partitions consiste alors à inverser l'attribut (dimension) courant sur la présentation donnée avec un autre attribut, qui fera l'objet d'une analyse ou d'une comparaison.

Definition 5.9 (Rotation)**.** *L'opérateur de rotation, appelé* rotate*, permet de substituer un attribut à un autre dans l'ensemble des attributs (ou schéma de résumé) qui décrivent les résumés d'une partition. Il est défini comme suit :* $P' = rotate(P, A_i, A_j)$*, où le schéma d'un résumé dans* P *est* (X, A_i) *et dans* P' *est* (X, A_j)*.*

Exemple 5.22 (Rotation sur une partition de résumés). *Soit le résumé* $z \in P$ *avec le schéma suivant* (A, B, C)*, où* B *et* C *correspondent respectivement aux dimensions ACTIVITE et REVENU,comme dans le tableau 5.5. Pour les besoins de l'analyse, l'utilisateur souhaite mettre en évidence la dimension ACTIVITE avec la dimension AGE notée* D*. L'opération de rotation qu'il effectue est la suivante :* $P' = rotate(P, C, D)$ *cela donne un résumé* $z' \in P'$ *avec un schéma* (A, B, D)*.*

Cet opérateur, permet donc de changer le schéma à la visualisation d'un résumé pour sa présentation au sein d'une partition. Cet opérateur agit sur le contenu intentionnel du résumé, vu qu'il manipule les attributs qui le décrivent et modifie son schéma. Nous proposons ci-dessous des opérateurs de restructuration dont le seul but est de réorganiser la présentation sans modification au niveau du schéma ou du contenu qu'il soit intentionnel ou extensionnel.

5.2.3.3 Opérations identité

L'objectif des opérations d'identité est de donner à l'utilisateur la possibilité de visualiser une partition de résumés, qui est un ensemble sans ordre, de manière organisée. L'application de ce genre d'opérateur produit une partition qui reste la même dans le contenu et dont seulement la présentation est modifiée. On note alors un tel opérateur d'identité $Id(P) = \langle P \rangle = Op(P, param)$, où Op est l'opérateur d'dentité à appliquer, et $param$ détermine les éléments essentiels pour établir un ordre. La notation $\langle P \rangle$ signifie alors que la partition reste la même sauf que l'opération appliquée permet de l'ordonner d'une certaine manière selon l'opérateur choisi. Nous proposons ici deux opérateurs le *switch* et le *sort*.

L'opérateur *switch* et l'opérateur *sort* de l'algèbre multidimensionnelle qui manipule les cubes de données, sont conçus pour aider l'utilisateur à construire la représentation tabulaire d'une partition de résumés. Ils produisent tous les deux la même partition de résumés utilisée en entrée, en adoptant des points de vue variés.

Opération de permutation

L'opération de permutation *switch* dans les systèmes OLAP est appliquée sur les valeurs de chaque dimension. Elle permet de changer la position de ces valeurs. L'intérêt de cette opération est qu'en réorganisant des dimensions elle permet de mettre en relief certaines valeurs de mesure, sur les partitions de résumés, la permutation consiste à contrôler les positions des descripteurs et des attributs dans le tableau.

Definition 5.10 (Switch, permutation)**.** *L'opérateur Switch est appliqué à une partition de résumés P pour inverser deux positions indiquées en paramètres. Nous le définissons comme :*
$\langle P \rangle = Switch(P, C, pos_i, pos_j)$*, où le critère* $C \in \{$*attribut, descripteur ou résumé*$\}$ *détermine la nature des éléments à permuter, et* pos_i, pos_j *sont les deux positions à inverser.*

Exemple 5.23 (Opération de permutation (*switch*)).
Considérons l'exemple précédent dans le tableau 5.5. L'utilisateur souhaite comparer des informations sur les individus décrits par les résumés de ce tableau (base de données des SIMPSONS tirée de [83]), qui sont hommes d'affaires ou artistes. Pour cela, il préfère visualiser ces informations l'une à côté de l'autre. Le tableau 5.6 illustre le résultat de l'opération switch sur l'attribut ACTIVITE entre la position du descripteur artiste et la position du descripteur agent de sécurité. L'opération s'écrit : $Switch(P_3, Desc, pos_2, pos_4)$.

L'opérateur *switch* peut s'appliquer sur chaque entrée de la table, c-à-d sur les résumés, sur les descripteurs ou sur les attributs. Il permute simplement les positions de deux éléments distincts.

α_d	ACTIVITE				REVENU		
	art	**h.aff**	ag.s	**emp**	mise	mod	enor
z_{11}	0.8	0.7			1.0		1.0
z_{121}			0.9		1.0		
z_{122}			0.3			1.0	
z_{13}			0.8				1.0
z_2	0.8					1.0	1.0
z_3		0.8					1.0

Table 5.6 – Permutation sur les positions des descripteurs pour l'attribut ACTIVITE dans P_3

Opération de tri

Nous pouvons également appliquer l'opération de tri (*Sort*), sur des partitions de résumés selon un ordre croissant ou décroissant (minimum ou maximum). Ce tri peut bien s'appliquer aux attributs ou aux descripteurs selon un ordre alphabétique ou encore aux valeurs contenues dans les cellules comme par exemple le degré d'appartenance ou la cardinalité.

Definition 5.11 (Sort, tri)**.** *L'opérateur sort est appliqué à une partition de résumés P pour établir un ordre sur un critère ou un paramètre de la partition. Il est noté :*

$$\langle P \rangle = SORT(P, C, ordre)$$

où, $C \in \{attribut,\ descripteur,\ cardinalités \ldots\}$ et ordre correspond à un ordre lexicographique ou un ordre naturel sur des réels, il peut être croissant ou décroissant.

Exemple 5.24 (L'opération de tri (*Sort*)).
Si on veut trier selon leurs degrés de satisfaction α l'ensemble des résumés de la partition P_3 sur le tableau 5.6, l'opération s' écrit : $\langle P_3 \rangle = SORT(P_3, \alpha_{agent.\ sécurité}, minimum)$, voir le résultat sur le tableau 5.7.
L'opérateur de tri peut également s'appliquer sur chacune des dimensions du tableau, suivant un ordre donné. Par exemple, il est possible de trier des résumés suivant les degrés de satisfaction sur un descripteur. Le tableau 5.7 montre un exemple, où on cherche à trier les résumés de P_3 selon les degrés de satisfaction du descripteur agent de sécurité suivant un ordre croissant.

5.2.4 Synthèse

Il est important de mentionner que quelques opérateurs de restructuration OLAP habituels pour les cubes de données, tels que le *push*, le *pull* ou *un/nest* n'ont pas été pris en considération dans cette section. Ces opérateurs n'ont aucune traduction possible dans notre modèle de partition de résumés. Sur le tableau 5.8,

α_d	ACTIVITE				REVENU		
	art	**h.aff**	ag.s	**emp**	mise	mod	enor
z_{122}			0.3			1.0	
z_{11}	0.8	0.7			1.0		1.0
z_{13}			0.8				1.0
z_2	0.8					1.0	1.0
z_3		0.8					1.0
z_{121}			0.9		1.0		

Table 5.7 – Tri de P_3 selon $\alpha_{agent\ de\ sécurité}$

nous avons fait un récapitulatif de tous les opérateurs définis jusqu'ici. Ils sont organisés en catégories. Nous donnons aussi une brève description et le rôle de chaque opérateur.

5.3 Á propos de l'algèbre

Nous allons nous intéresser dans cette section à compléter quelques caractéristiques liées à l'algèbre définie dans ce chapitre. Nous éclaircissons tout d'abord la différence existante entre notre algèbre et l'algèbre possibiliste, ensuite nous consacrons le reste de cette section à étudier la fermeture et la sémantique de l'algèbre.

5.3.1 Modèle conjonctif.

La structure multidimensionnelle que les opérations algébriques définies dans ce chapitre manipulent est basée sur les partitions de résumés flous utilisant des concepts vagues. Ce modèle considère pour chaque attribut, une sémantique conjonctive. C'est à dire que si nous prenons un individu ou un concept, présenté sur l'attribut `AGE` par les descriptions (jeune, âgé). Dans notre modèle, nous considérons deux étiquettes pour décrire l'attribut `AGE`, c'est à dire par `jeune` **et** `âgé`.

D'un autre côté, il existe des travaux qui ont proposé l'extension de l'algèbre relationnelle aux bases de données possibilistes dans [81]. Ils prennent en considération, dans cette extension, les valeurs disjonctives d'un attribut. Ce qui signifie que sur l'attribut `AGE`, la distribution possibiliste sera décrite par l'une des deux étiquettes, `jeune` **ou** `âgé`.

Ce qu'il faut mentionné ici, c'est que la sémantique associée aux opérateurs algébriques définies sur le modèle des partitions de résumés flous, n'est pas similaire aux opérateurs définis sur les modèles possibilistes. La différence réside dans le fait d'utiliser des concepts sous forme conjonctive dans notre modèle et disjonctive dans le modèle possibiliste.

CATEGORIE	OPERATION	DESCRIPTION	
Classique	**Slice**	Syntaxe	$P' = \sigma(P, z\theta\tilde{v})$.
		Rôle	Sélection sur les valeurs d'attributs.
	Dice	Syntaxe	$P' = \sigma(P, pred(z))$.
		Rôle	Sélection sur les propriétés des résumés.
	Projection	Syntaxe	$P' = \Pi_{A_1,..,A_k}(P)$.
		Rôle	Réduction du nombre d'attributs dans une partition.
	Fusion	Syntaxe	$P' = merge(P, \{z_1, \ldots, z_k\})$.
		Rôle	Regroupement des résumes par agrégation.
	Produit Cartésien	Syntaxe	$P'' = P \times P'$.
		Rôle	Combinaison des résumés d'une partition avec une autre.
	Jointure	Syntaxe	$P^* = P \underset{pred(z.A,z'.A')}{\bowtie} P'$
		Rôle	Connexion de deux partitions selon un critère. Filtrage d'un produit cartésien.
Ensembliste	**Union**	Syntaxe	$P'' = \{z \mid z \in P \vee \ z \in P'\}$.
		Rôle	Union des résumés dans P et P'.
	Intersection	Syntaxe	$P'' = \{z \mid \ z \in P \ \wedge z \in P'\}$.
		Rôle	Intersection des résumés dans P et P'.
	Différence	Syntaxe	$P'' = \{z \mid z \in P \wedge z \notin P'\}$.
		Rôle	Différence des résumés dans P et P'.
Granularité	**Drill-down**	Syntaxe	P' = Drill-down $(P = \{z_1, z_2, \ldots, z_n\}, z_1)$, $P' = \{z'_1 \ldots z'_n, z_2, \ldots, z_n\}\}$
		Rôle	Spécialisation d'une partition sur un résumé.
	Roll-up	Syntaxe	P' = Roll-up $(P = \{\{z_1, \ldots, z_i, z_{i+1}, \ldots, z_n\}, \{z_1, \ldots, z_i\})$ $P' = \{z', z_{i+1}, \ldots, z_n\}$
		Rôle	Généralisation d'une partition sur un sous-ensemble de résumés.
Restructuration	**Rotation**	Syntaxe	$\langle P \rangle = rotate(P, A_i, A_j)$.
		Rôle	Inversion des deux attributs dans la partition P.
	Permutation	Syntaxe	$\langle P \rangle = Switch(P, C, pos_i, pos_j)$
		Rôle	Modification de l'ordre des paramètres sur les résumés.
	Tri	Syntaxe	$\langle P \rangle = SORT(P, C, ordre)$
		Rôle	Tri de P selon le critère C.

Table 5.8 – Tableau récapitulatif des opérations définies dans l'algèbre de manipulation des partitions de résumés

5.3.2 Composition et fermeture.

Chacune des opérations définies dans l'algèbre considère au moins une partition de résumés en entrée. En revanche, et tout en respectant les conditions de l'opérateur utilisé, chaque opération produit en sortie une partition de résumés. Ainsi, l'algèbre proposée pour manipuler les partitions de résumés est une algèbre fermée. De plus, la partition résultat elle-même peut faire l'objet de l'application d'un autre opérateur. Cela signifie que la composition d'opérateurs (par exemple $\sigma\ \rho\ \pi$) est valide pour notre algèbre.

La complétude de l'algèbre de manipulation définie dans ce chapitre, signifie que toute manipulation pouvant être souhaitée par l'utilisateur devrait pouvoir être exprimable par une expression algébrique.

Grâce à la fermeture de l'algèbre, nous pouvons déduire que toutes les opérations peuvent être utilisées dans une combinaison pour exprimer une requête complexe. Ceci est dû au fait, que toutes les opérations produisent en résultat une partition de résumés, comme déjà indiqué, cette partition peut faire l'objet d'une entrée pour une nouvelle opération.

Nous définissons une expression algébrique en utilisant notre algèbre. Une expression utilise un ou plusieurs opérateurs de l'algèbre avec une ou plusieurs partitions comme opérandes. Soit E l'ensemble de toutes les expressions possibles, avec $e \in E$ une expression algébrique. Si l'expression e est composée d'un ensemble d'opérateurs algébriques, soit par exemple $e = op_1\ \rho\ op_2$. Il est à noter que la relation ρ existant entre deux opérations est une *relation binaire*, ce qui implique qu'elle est réflexive, symétrique et transitive.

5.3.3 Sémantique de l'algèbre

Dans cette section, nous étudions la sémantique de l'algèbre définie dans ce chapitre. Pour cela nous nous sommes intéressés à quelques critères usuels propres à valider le calcul proposé. Les critères retenus sont les suivants :

1. *La signification des opérateurs.* Chaque opération définie dans l'algèbre d'une manière syntaxique, a une sémantique vis-à-vis du résultat qu'elle fournit. Cette sémantique est issue de l'algèbre relationnelle pour les opérateurs classiques, et de l'algèbre des cubes multidimensionnels pour les opérateurs de granularité et de restructuration. Au cours des définitions proposées, chacune des opérations a été illustrée par un exemple afin de montrer son utilité. La catégorisation adoptée pour présenter les opérateurs fait également sens du point de vue de l'objectif général de chacune des opérations. De plus, un ensemble d'opérateurs de restructuration n'a pas été défini, pour la bonne raison que leur adaptation n'a pas trouvé de sémantique valable dans l'algèbre de manipulation des partitions de résumés flous.
2. *Le rapport de sens entre les opérateurs.* Ce critère est basé sur la fermeture et la composition de l'algèbre abordées dans la section 5.3.2. Du point de vue

sémantique, l'enchaînement d'un ensemble d'opérateurs doit permettre la réalisation de l'objectif d'analyse fixé par l'utilisateur. En effet, ce dernier compose les suites d'opérations dont il a besoin pour avoir une certaine partition sur laquelle il se pose une certain nombre de questions. La fermeture de l'algèbre permet donc de conserver la sémantique des opérateurs, qu'ils soient utilisés seuls ou dans une expression avec plusieurs autres opérateurs.

3. *Les conditions de vérité d'une opération.* En définissant les opérations de l'algèbre de manipulation des partitions de résumés, nous avons été parfois amené à poser des conditions pour respecter la fermeture de l'algèbre. Alors, pour ceci nous avons exigé pour chaque partition résultante de l'application d'une opération, qu'ils n'y aient pas de résumés conflictuels. Une autre condition a été posée dans le cadre des opérations binaires, elle consiste à ne prendre que les partitions ayant une intersection vide. Les conditions de vérité des opérations définies respectent toutes la propriété de faible orthogonalité et de complétude des partitions manipulées et produites, afin de garantir la fermeture de l'algèbre.

Nous avons aussi, souligné pour chacune des opérations, l'impact de son application sur l'extension et la cardinalité des résumés manipulés au sein d'une partition.

La définition des contraintes pour l'application d'un certain nombre d'opérations nous interdit des situations d'utilisation d'opérateurs comme pour l'exemple de la projection conflictuelle. Ce type d'interdiction ne nous permet pas de prétendre la complétude de l'algèbre, en revanche il nous garantit la fermeture.

La sémantique de la composition des opérateurs dépend donc des conditions de la fermeture de l'algèbre. Nous avons expliqué dans le chapitre précédent qu'une coupe de la hiérarchie est une partition qui respecte non seulement la propriété de faible orthogonalité mais aussi la complétude (la couverture de la relation initial R). La fermeture de l'algèbre proposée est garantie pour l'ensemble des partitions. Sachant que le point de départ de toute manipulation dans notre modèle est une coupe, nous allons associer chaque partition P à une coupe C de le hiérarchie par une relation (C, P). Ainsi une partition qui est résultat par exemple de l'application d'une opération de sélection ou d'une projection peut ne pas être une coupe de la relation mais le fait d'avoir une coupe à laquelle elle est associée nous permet de pouvoir lui appliquer des opérateurs comme le *roll-up* ou le *drill-down* qui agissent sur les coupes de la hiérarchie.

Nous rappelons, que l'objectif de cette algèbre et de fournir à l'utilisateur un outil qui permet de le guider pour la manipulation et l'analyse des résumés flous. La finalité de cette algèbre est de supporter un outil convivial et intuitif pour guider l'utilisateur dans ses analyses. Nous proposons pour ceci le passage d'une forme d'expression algébrique à une interface visuelle de formulation qui décharge l'utilisateur de toute connaissance de la syntaxe de notre algèbre. Cette interface orientée utilisateur est présentée dans la section suivante.

5.4 Interface graphique

Nous avons présenté jusqu'ici un modèle de données ainsi qu'une algèbre pour manipuler des résumés linguistiques de données relationnelles. Nous avons donné une structure multidimensionnelle à ces résumés pour permettre leur manipulation. L'objectif est de fournir à l'utilisateur un outil d'exploration ainsi que la possibilité d'interroger et de manipuler des partitions de résumés flous, issues d'une hiérarchie générée par le système SaintEtiQ.

Nous présentons dans cette section l'interface d'exploration interactive des résumés linguistiques flous. Nous présentons tout d'abord, les données factuelles sur l'implémentation de cette interface. Ensuite l'interface graphique orientée utilisateur est présentée, en suivant quelques exemples de fonctionnalités qu'elle offre pour la manipulation d'une partition de résumés.

5.4.1 Implémentation

Afin de valider le modèle proposé et son algèbre de manipulation et de visualisation, nous proposons d'implémenter une interface de représentation et de manipulation des résumés. Nous avons utilisé pour la mise en place de cette interface différents outils :

1. Les résumés sont stockés dans des documents XML. En effet, les résumés générés par SaintEtiQ sont présentés sous forme de documents. Chaque résumé évolue au cours de l'apprentissage et contient la référence aux résumés fils sous forme de liens. Chacun de ses documents possède une syntaxe XML spécifiée dans un formalisme *XML-Schema* [2].

2. L'extraction des résumés ainsi que les informations nécessaires pour l'application de l'algèbre est gérée par un analyseur syntaxique (*parser*) XML.

3. Les opérateurs de manipulation des résumés sont implémentés avec le langage de programmation JAVA sous l'environnement de programmation *netBeans4.1.*

La hiérarchie des résumés est stockée dans un document XML, contenant les informations sur chaque nœud, représentant un résumé. La figure 5.2 montre un exemple de résumé, sérialisé dans un document XML. Ce document est construit sur la base de la définition du résumé telle qu'elle a été introduite dans les chapitres précédents et consiste à définir un résumé selon le triplet $(z = (I_z, E_z, R_z))$ de son intention, son extension et la relation qui existe avec ses nœuds fils.

Les données d'entrée sont stockées dans deux fichiers XML : le premier est celui qui contient la hiérarchie des résumés générés par SaintEtiQ, et le second est celui qui contient les connaissances de domaine (BK) ayant servi pour la construction des résumés.

[2] les détails de ces schémas sont fournis dans [90].

```
<Summary HierarchyID="2" summaryID="5"
         parentID="4" cardinality="2" relativeCard="3/2"
         similarity="0.87" xlink:type="simple" xlink:href="SampleHierarchy/3"
         xmlns="http://www.simulation.fr/seq/SummaryDefinition.xsd"
         xmlns:xlink="http://www.w3.org/1999/xlink"
>
    <Intention>
        <Attribute name="productName" domainCard="5">
            <Descriptor name="beverage" sat="0.4" support="1/2" />
            <Descriptor name="food" sat="1" support="1" />
        </Attribute>
        <Attribute name="price" domainCard="5">
            <Descriptor name="average" sat="1" support="1" />
            <Descriptor name="cheap" sat="1" support="1/2" />
        </Attribute>
        <Attribute name="packaging" domainCard="5">
            <Descriptor name="box" sat="1" support="1" />
            <Descriptor name="can" sat="1" support="1/2" />
        </Attribute>
    </Intention>
    <Extension>
        <TupleID relativeCard="1">0005</TupleID>
        <TupleID relativeCard="1/2">0013</TupleID>
        <TupleID relativeCard="1">0042</TupleID>
    </Extension>
    <Edges count="3" />
      <Child xlink:type="simple" xlink:href="2/12" />
      <Child xlink:type="simple" xlink:href="2/17" />
      <Child xlink:type="simple" xlink:href="2/33" />
    </Edges>
</Summary>
```

Figure 5.2 – Exemple de document résumé

5.4.2 Les fonctionnalités

L'objectif de cette interface est de fournir un outil adapté à l'utilisateur final en élaborant des fenêtres simples à interpréter et en proposant des contrôles intuitifs pour un utilisateur lambda. Ainsi les principaux éléments de l'interface sont :

- La visualisation de la hiérarchie générale.
- La visualisation des partitions sous forme tabulaire.
- L'application d'opérateurs de manipulation sur les partitions avec la possibilité de préciser les résumés concernés par cette manipulation.
- La possibilité de choisir un attribut bien précis et des descripteurs spécifiques, pour explorer une partition.

Selon le modèle de partition proposé dans le chapitre précédent, la première fonctionnalité de cette interface est la présentation d'une liste de partitions calculées à partir de la hiérarchie ainsi que d'établir un ordre total existant sur cette liste.

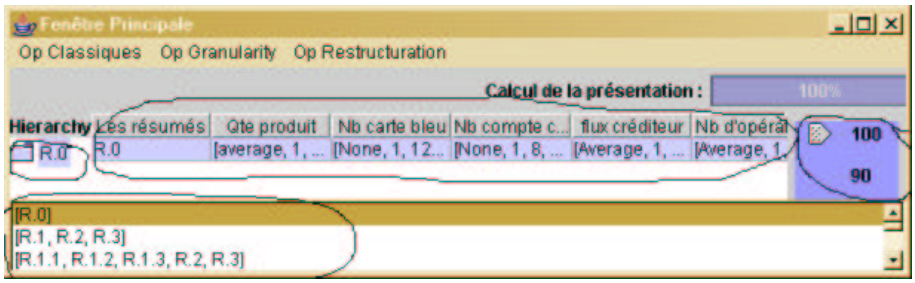


Figure 5.3 – Fenêtre d'accueil de l'interface graphique

La figure 5.3, montre la fenêtre d'accueil de l'interface graphique utilisée. Cette fenêtre présente la hiérarchie de résumés, leurs niveaux d'abstraction ainsi que la liste des partitions construites sur cette hiérarchie.

Le niveau d'abstraction. L'utilisateur a la possibilité de sélectionner sur l'ensemble de la relation R, un niveau d'abstraction avec lequel il souhaite manipuler la partition de résumés correspondante, qui fera l'objet de son analyse. Ainsi sur la fenêtre d'accueil est affiché un *curseur* [3] qui permet de choisir un taux de compression pour la relation originale dans la liste des partitions pré-calculées. Le *curseur* offre une graduation sur une échelle allant de 0 à 100%, pour présenter le taux de compression d'une partition par rapport à la base originale.

Sur ce curseur, le taux de compression maximal représente le nœud racine de la hiérarchie ce qui correspond au degré de généralisation le plus élevé de la hiérarchie. Notre choix, seulement par simplification, est que le curseur soit positionné par défaut sur ce degré maximal.

[3] traduction en français d'un slider.

5.4.3 Exploration

Cette interface est principalement construite pour intégrer des fonctionnalités de manipulation et de présentation de partitions de résumés déjà définies dans ce document.

La fenêtre d'accueil de l'interface utilisateur, présentée à la figure 5.3, permet de sélectionner la partition à explorer, selon un niveau d'abstraction donné. Cette partition fera l'objet de différentes manipulations.

5.4.3.1 La visualisation

La visualisation de la partition choisie, consiste à représenter les résumés de cette partition sous une forme tabulaire sur l'ensemble des attributs concernés. La figure 5.4, montre cette fenêtre d'affichage d'une partition.

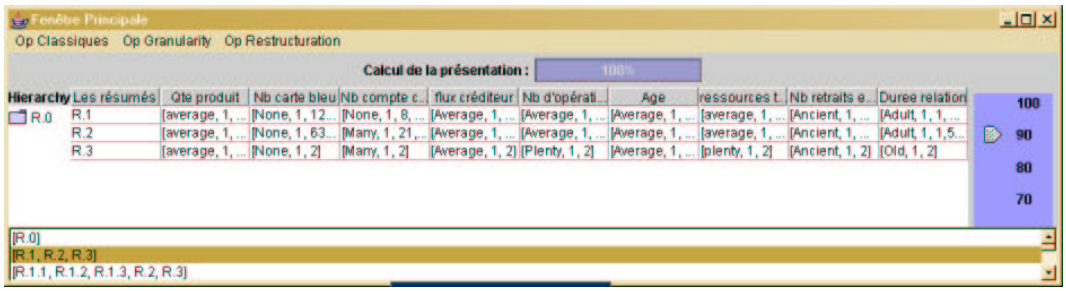


Figure 5.4 – Fenêtre d'affichage d'une partition de résumés

Les informations sont affichées selon une représentation tabulaire, ce qui est particulièrement adapté à l'utilisateur pour sélectionner le résumé qu'il veut détailler ou généraliser. Dans la représentation d'une partition, pour chaque résumé les informations suivantes seront fournies :

- l'ensemble des étiquettes floues représentant le résumé sur chaque attribut, comme le montre la figure 5.5.
- les différentes informations du résumé, il suffit que l'utilisateur clique sur le résumé qui fera l'objet de son interrogation pour afficher les informations concernant ce résumé (cardinalité, nombre de fils, taille de l'extension ...), voir la figure 5.5.
- la possibilité de zoomer sur la partition suivant le résumé choisi.

5.4.3.2 Les opérateurs

Le menu déroulant de la figure 5.5 permet d'appliquer les différentes opérations possibles sur les partitions. Nous présentons ici un exemple des opérateurs de granularité et de sélection.

Dans un processus d'analyse et d'exploration, l'utilisateur est amené à passer par plusieurs partitions de résumés intermédiaires avant d'atteindre son objectif. Si par exemple l'utilisateur veut afficher les détails d'un résumé choisi dans une

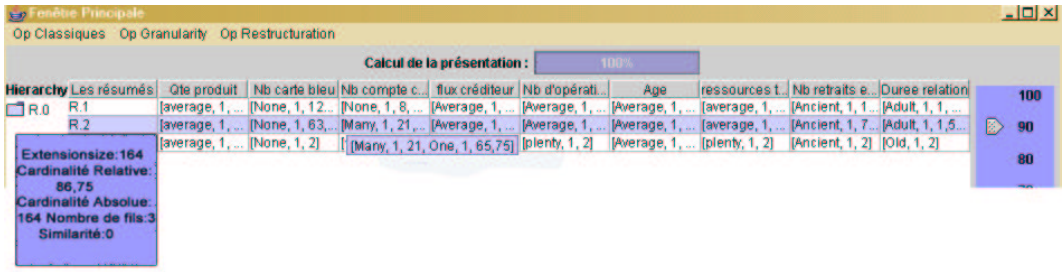


Figure 5.5 – Informations sur le résumé et sur les attributs

partition il est amené à faire varier la granularité de la partition sur ce résumé. Les opérateurs de granularité sont appliqués directement sur la partition du résumé ou sur le résumé lui même au moyen d'une simple manipulation de la souris. Par exemple, la manipulation concernant le raffinement (drill-down) du résumé permet de détailler les résumés à un niveau de détail plus fin, sachant que les résumés sont liés par une relation d'ordre. Ce procédé de raffinement contribue à l'aide à l'analyse par la navigation, étant ainsi possible de parcourir les différents niveaux de la hiérarchie.

Les opérateurs de granularité notamment le drill-down et le roll-up présentés respectivement dans l'interface par zoom-in et zoom-out ont un impact direct sur la visualisation d'une partition de résumés. En effet, ces deux opérateurs permettent de "zoomer" sur la partition de résumés courante. Leur application permet d'afficher la partition mère (zoom-out) ou fille (zoom-in) immédiate du résumé choisi, et d'afficher la partition à l'écran avec le niveau approprié matérialisé par la position du curseur.

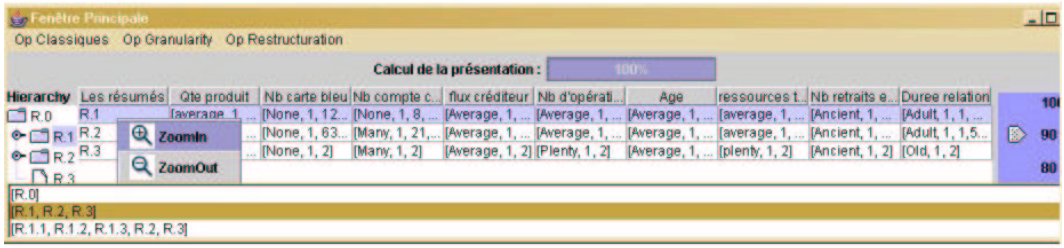


Figure 5.6 – Application du zoom-in sur le résumé $R.1$

Les figures 5.6 et 5.7, montrent un exemple de l'application d'un zoom-in (équivalent à un drill-down) sur le résumé $R.1$ de la partition $P = \{R.1, R.2, R.3\}$. L'application de l'opérateur drill-down sur $R.1$ permet d'afficher la partition $P' = \{R.1.1, R.1.2, R.1.3, R.2, R.3\}$ contenant les nœuds fils (résumés) de $R.1$. La fenêtre affiche tous les détails de chaque résumé sur chaque attribut, le taux de compression de la partition P' (ici égal à 80%) ainsi que la position des résumés résultats sur la hiérarchie.

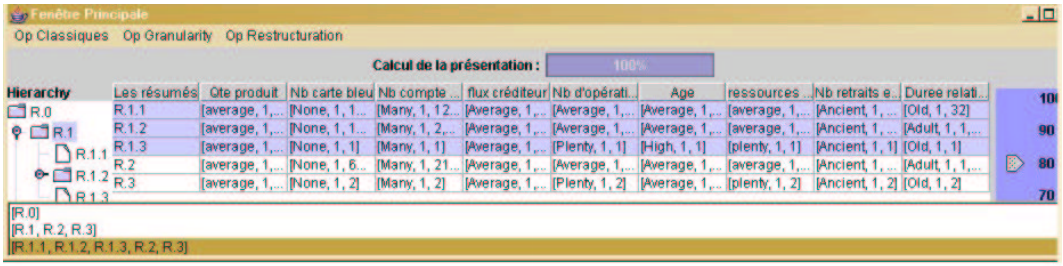


Figure 5.7 – Résultat du zoom-in sur le résumé $R.1$

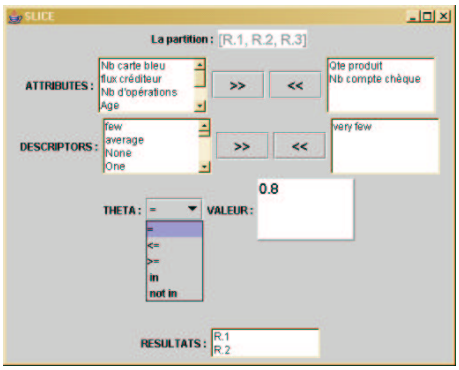


Figure 5.8 – Fenêtre de définition des critères de sélection

L'interface nous permet également l'interrogation d'une partition en appliquant des opérateurs de manipulation classiques, tels qu'ils ont été définis précédemment. Nous choisissons de montrer l'exemple de la fenêtre permettant de saisir une sélection sur les résumés *slice* d'une partition, en spécifiant différents critères sur les attributs et sur les valeurs recherchées. La figure 5.8, illustre cette opération.

Ci-dessous, est présenté un scénario d'une analyse, avec l'utilisation de quelques opérations de l'algèbre définie.

Scénario orienté prise de décision

L'exploration d'une hiérarchie, comme celle de l'exemple 5.1, permet à l'utilisateur d'avoir des informations d'ordre général. Les requêtes nécessitant des résultats précis ne peuvent avoir des réponses par une simple navigation exploratoire de la hiérarchie. L'algèbre proposée précédemment permet en effet à un utilisateur d'explorer les résumés en appliquant des manipulations dans un but analytique.

Nous prenons en considération un extrait d'une base de données sur des individus clients d'une banque afin d'étudier leur comportement vis-à-vis des produits proposés. La base est une relation dans laquelle chaque enregistrement repré-

sente un client, et où les attributs décrivent le client par des termes sur son âge, son revenu ou son activité, ainsi que les produits bancaires que le client utilise (comptes, cartes de crédit, prêts, ...).

Nous supposons que l'utilisateur souhaite analyser l'utilisation d'un client pour l'ensemble des services bancaires, et singulièrement *il cherche à savoir s'il existe une relation entre la fidélité d'un client et le niveau de son revenu.*

Pour ceci, l'utilisateur sélectionne tout d'abord le niveau de granularité avec lequel il souhaite travailler. Ce niveau doit lui fournir une vue complète de la relation qui peut être explorée en utilisant la liste des résumés de ce niveau. Ensuite, il spécifie la valeur de la durée qu'il désire conserver pour décrire la fidélité, cette durée étant décrite par les étiquettes linguistiques ancien, moyen ou récent. Pour exprimer la fidélité d'un client, l'utilisateur choisira l'attribut *durée* avec le descripteur *ancien*. Seuls les résumés représentés par ce descripteur seront alors sélectionnés. Sur l'ensemble de ces résumés, il pourra préciser un seuil de fidélité sur le degré de satisfaction du descripteur *ancien*, afin de ne garder que les résumés contenants les clients qu'il considère comme fortement fidèles.

L'utilisateur peut dans l'étape suivante raffiner sa requête. Pour ceci, supposons qu'il souhaite étudier dans le détail les résumés retenus afin de mieux expliquer la fidélité des clients. Pour ceci il cherche à visualiser la partition qui contient ces résumés à un niveau de granularité plus élevé. Ensuite, il ne gardera que les résumés ayant un certain revenu.

Si nous considérons que P est la partition en entrée de ce scénario. L'expression algébrique que l'utilisateur génère pour une telle requête considérée comme complexe, est la suivante :

$$P' = \pi_{durée,revenu}(\sigma(P,\ z.durée(ancien) \geq s)$$

où s est un seuil fixé. Finalement, une simple opération de forage vers le bas permet de donner les détails de chaque résumé que l'utilisateur souhaite analyser.

Le scénario descriptif qui vient d'être présenté montre la façon dont les opérateurs de l'algèbre d'analyse en ligne définies dans ce chapitre peuvent aider un utilisateur dans l'analyse des données représentées par les résumés d'une hiérarchie.

5.5 Conclusion

Dans ce chapitre, nous avons défini une collection d'opérateurs algébriques sur l'espace multidimensionnel de partitions de résumés. Ces opérateurs sont à la base d'une algèbre de manipulation des résumés. Cette algèbre prend en compte les spécificités du modèle de résumé que nous traitons et permet d'explorer succinctement le contenu de la base de données sans y accéder. Nous avons adapté la majorité des opérateurs d'analyse proposés dans les systèmes OLAP. Ainsi, nous avons identifié pour cette algèbre trois catégories d'opérateurs : les opérateurs de base (restriction, projection, union, ...) issus de l'algèbre relationnelle, les

opérateurs de changement de granularité (*roll-up* et *le drill-down*) et les opérateurs de restructuration (rotation, permutation, ...). L'aspect flou des résumés a été un élément principal pour l'adaptation des opérateurs d'analyse en ligne aux résumés linguistiques de SaintEtiQ.

Nous nous intéressons dans le chapitre suivant à la représentation des résumés et des partitions de résumés produits par SaintEtiQ, notamment pour en fournir une présentation claire et facile à interpréter par l'utilisateur final.

Chapitre 6

Représentation des résumés par prototypes flous

L'art ne veut pas la représentation d'une chose belle mais la belle représentation d'une chose.

— Emmanuel, Kant, .

6.1 Introduction

Généralement, la caractérisation par prototypes flous permet de fournir une description d'un ensemble de données dans l'objectif de rendre facile l'interprétation de ces données par l'utilisateur. La représentation prototypique réalise une mise en correspondance de classes identifiées avec des concepts naturels utilisés intuitivement pour décrire les données.

Dans les chapitres précédents, le modèle SaintEtiQ [83] montre que les résumés linguistiques constituent un moyen privilégié pour appréhender le contenu d'une base de données. Grâce aux caractérisations linguistiques fournies par des connaissances de domaine, ce modèle présente l'avantage d'adopter une représentation intelligible des résumés. Leur construction est incrémentale et réalise une classification hiérarchique des données vues à travers ces connaissances de domaine. Cependant, une fois les résumés obtenus, se pose le problème de leur analyse pour un utilisateur désirant extraire de l'information. Bien que l'interprétation d'un résumé soit simple elle devient difficilement envisageable pour un nombre élevé de résumés. Nous proposons donc d'enrichir la hiérarchie de résumés produits par SaintEtiQ, dans le cadre de la représentation prototypique des sous-ensembles flous.

L'objectif de ce chapitre est de caractériser les résumés, en définissant un représentant typique des n-uplets contenus dans le résumé. Cette représentation tire profit du cadre formel de la théorie des sous-ensembles flous, pour modéliser les limites imprécises du prototype. Dans ce contexte, trois méthodes différentes de construction de prototype ont été étudiés, pour chaque vue ou coupe de la hiérarchie des résumés. Le calcul de prototype que nous présentons cherche à

trouver la meilleure façon de fournir à l'utilisateur une interprétation purement linguistique des résumés.

6.1.1 Problématique

Considérons à un niveau d'abstraction donné de la hiérarchie produite par SaintEtiQ une partition contenant un grand nombre de résumés. Chacun des résumés constitue une forme de connaissance sur les données résumées, cette connaissance est représentée par l'intention du résumé décrivant les n-uplets se trouvant dans son extension. Toutefois, l'information contenue dans la description intentionnelle d'un résumé est considérée comme une représentation difficile à interpréter par un utilisateur, étant donné l'utilisation d'un ensemble de descripteurs chacun avec un degré de satisfaction sur chaque attribut. Cette structure issue de la théorie des sous-ensembles flous est certes efficace pour souligner les limites de l'incertitude mais elle contribue à la complexité de la représentation du modèle de données, défini dans un chapitre précédent, représenté par une partition de résumés flous. Il est donc légitime d'étudier de quelle manière représenter les résumés au sein d'une partition afin de faciliter l'interprétation par l'utilisateur de l'information qu'il peut trouver dans un tel modèle.

Nous proposons dans la suite de ce chapitre d'étudier la construction des prototypes flous à partir des résumés d'une partition, dans l'objectif de les présenter à l'utilisateur. L'idée générale de cette proposition est de disposer d'une description concise en langage naturel sous forme d'un prototype qui représente un résumé à l'aide d'un unique descripteur linguistique sur chaque attribut.

6.2 Les prototypes flous

Le terme prototype désigne un élément choisi pour représenter un ensemble de données : il est unique et caractérise l'ensemble de ces données, les résume et met en avant les éléments les plus importants, facilitant en cela l'interprétation par l'utilisateur.

De manière générale on appelle " prototype " l'élément choisi pour résumer un groupe de données résultant par exemple d'une étape de clustering : c'est un unique individu qui caractérise le groupe, l'ensemble des prototypes peut ensuite être utilisé comme une représentation simplifiée de l'ensemble initial des données.

Dans la représentation d'un prototype, on utilise le plus souvent, un point unique associé au groupe. Ce point est considéré comme le centre du groupe, il peut être calculé selon diverses méthodes (moyenne du cluster, moyenne pondérée, médiane ou autres). Dans ce chapitre nous envisageons de trouver une représentation pour un résumé de données considéré comme un sous-sensemble flou. Le représentant de ce résumé ne peut être réduit à un point, d'où le caractère flou du prototype à construire. Ce prototype est donc considéré comme un concept flou mais à une condition que sur chaque attribut il n'y aura qu'un descripteur linguistique. Le caractère flou des prototypes des résumés est aussi justifié par

l'utilisation des étiquettes linguistiques qui expriment une notion imprécise d'un individu décrit sur un attribut. D'autres approches existantes peuvent associer une représentation plus riche que des points comme les méthodes de clustering flou qui fournissent des sous-ensembles flous représentant un groupe mais qui ne correspondent pas à des prototypes, parce que souvent ces représentants donnent une vision très simple en modélisant des points qui appartiennent simultanément à plusieurs clusters.

Notre intérêt pour la représentation des résumés flous s'inscrit dans le cadre de la représentation des concepts flous. Un résumé tel qu'il est a une représentation intentionnelle qui est une description synthétique des éléments appartenant à l'extension du résumé. Sur chaque attribut A, un concept vague $z.A$ est défini, qui correspond à toutes les valeurs possibles que peut prendre un n-uplet de R_z sur A. Les sciences cognitives se sont intéressées à la représentation des concepts flous. En effet, les sciences cognitives s'intéressant à la représentation cognitives des catégories ont montré que certains éléments d'une catégorie peuvent être considérés comme les plus typiques et constituent donc les meilleurs exemples ou représentants de la catégorie. Ces travaux initié par E. Rosch [88], ont montré que les membres d'une catégorie ne sont pas tous équivalents, quelques uns étant plus représentatifs ou typiques que d'autres. Ils ont montré aussi qu'un point est typique s'il est similaire aux autres membres de son groupe et différent des membres d'autres groupes.

6.2.1 Travaux connexes

Plusieurs travaux se sont intéressés à étudier les prototypes flous et leur application. Dans les travaux de Frigui et Nasraoui [28], une approche de construction de prototype par sélection d'attribut pertinent a été définie. Les attributs ne sont pas déterminés globalement pour la totalité des données mais dépendent plutôt des groupes considérés et permettent de caractériser chacun des groupes. D'autres travaux [61, 60], ont proposé une approche pour la construction de prototypes flous. Dans ces travaux, un prototype flou est défini comme une agrégation des éléments les plus typiques. Il accentue les points communs des membres d'une catégorie mais également leurs caractéristiques distinctives. Il se fonde sur la notion de typicité qui modélise le fait que tous les membres d'un groupe ne représentent pas d'une manière égale le groupe de données. Cette notion de typicité d'un élément au sein d'un groupe dépend de deux mesures de comparaison : sa ressemblance et sa différence avec les autres éléments. Ces travaux se sont basés sur une procédure définie par Rifqi dans [87], qui construit des prototypes flous dans le cas où les données considérées sont floues. Les prototypes flous ont aussi été utilisés pour la découverte et la représentation des connaissances par Jose A. Olivas dans [77]. Il a définit le modèle *FPKD* pour *Fuzzy Prototypical Knowledge Discovery*, en se basant sur la définition d'un prototype proposée par Zadeh dans [104]. Zadeh dans son approche suggère de présenter un concept par un ensemble de prototypes représentant la compatibilité avec les échantillons du

concept par (haut, moyen ou bas). Il a aussi proposé récemment la notion de *protoform* [106], qui a été utilisée dans certaines approches de construction de résumés linguistiques [48, 47, 108].

6.2.2 Prototypes flous pour les résumés linguistiques

L.A Zadeh a proposé dans [106], la notion de *protoform* pour *forme prototypique*, qui est une forme générale d'un résumé de données linguistique. Il est défini comme un sigma-résumé, c'est-à-dire, un résumé de résumés. Par exemple, le protoform de la proposition " la plupart des Français sont grands " est " Q A est B", où Q est un quantificateur flou, et A, B sont des étiquettes floues. Dans [106], cette notion est utilisée pour les moteurs de recherche. Elle semble être une idée conceptuelle très puissante pour la formalisation d'un raisonnement cohérent humain, et pour les capacités de déduction dans les moteurs de recherche.

La notion de protoform, a été utilisée pour les résumés linguistiques dans l'approche de Kacprzyk [48]. Les résumés produits ont été utilisés pour construire des protoforms dans un processus de découverte de connaissances dans les travaux de Kacprzyk [48, 47, 108]. Dans ces travaux les auteurs travaillent sur des données linguistiques dans des résumés illustrés par *la plupart des employés sont jeunes et bien payés*. Ils considèrent que l'utilisation de la logique floue qui facilite l'utilisation du langage humain, cohérent et naturel fait la force d'un outil de découverte de connaissance. Et ils présentent, une extension d'une approche interactive, basée sur la logique floue et les requêtes exprimées par le système FQUERY (présenté dans le chapitre 3) pour les données floues [46, 107], pour impliquer des critères plus sophistiqués dans les méthodes de recherche. Ils recommandent pour ceci l'usage du concept de protoform, comme la forme générale d'un résumé de données linguistique, ainsi que la construction d'une hiérarchie de protoforms pour les résumés. Etant donné que le système SaintEtiQ s'inscrit dans la classe d'approches pour la construction des résumés linguistiques, nous avons pensé qu'il est intéressant d'étudier l'adaptation de la notion de prototype flou aux résumés de SaintEtiQ. Cette adaptation sera utilisée dans la représentation prototypique des résumés.

Dans ce qui suit notre objectif est de présenter à l'utilisateur final des résumés qui seront facilement interprétable, en s'appuyant sur la définition introduite par E. Rosch dans le cadre des sciences cognitives.

6.3 Prototypes de résumés

Dans cette section nous proposons une voie pour présenter les informations brutes que nous trouvons dans les résumés de données à l'aide de prototypes flous. Pour ceci nous étudions trois méthodes de construction de prototypes dans le contexte de partition de résumés. Les trois méthodes que nous proposons considèrent un résumé z dans une partition de résumés obtenue suivant une coupe de

la hiérarchie produite par SaintEtiQ, et sélectionnée par l'utilisateur. Ces trois méthodes sont les suivantes :

1. La première méthode, dite *idéale* consiste en la construction d'un prototype *idéalisé* à partir d'un résumé z, en analysant sa description sur chaque attribut. Cette méthode est considérée comme une approche multidimensionnelle, elle est sensible aux corrélations existantes entre les attributs, c'est une approche qui considère le résumé du point de vue de son intention.
2. La deuxième méthode, dite *réelle* consiste en la construction du prototype à partir des feuilles du résumé z qui sont les plus représentatives, afin de trouver celle qui représente le mieux le résumé. Cette seconde méthode est une approche qui se focalise sur la comparaison de n-uplets présents dans l'extension du résumé.
3. La troisième méthode, dite *hybride* consiste à trouver les représentants du résumé matérialisés par les feuilles du sous-arbre de z étant les plus proches du prototype dit *idéalisé*.

L'objectif de ces méthodes est celui de donner une représentation des résumés par le biais de prototypes flous. Le prototype fourni doit respecter la propriété de l'unicité. Nous posons légitimement la condition que le prototype que nous cherchons à construire est unique, vu que la facilité d'interprétation que nous souhaitons offrir à un utilisateur est celle d'associer chaque résumé à un unique représentant. Dans ce qui suit nous considérons un prototype par résumé de données. La suite de ce chapitre est consacrée à étudier les détails de chacune des méthodes et à discuter les avantages et les inconvénients de chacune.

6.3.1 Prototype idéalisé d'un résumé

Nous considérons ici l'intention du résumé qui offre un sous-ensemble flou sur chacun des attributs. Notre objectif est de trouver attribut par attribut le descripteur linguistique le plus représentatif du sous-ensemble flou $z.A_i$. Pour ceci nous définissons le *prototype idéalisé* sur l'intention du résumé.

Definition 6.1 (Prototype idéalisé d'un résumé)**.** *Le prototype idéalisé PrI(z) d'un résumé z est un élément du produit cartésien des étiquettes linguistiques qui decrivent chacun des attributs réécrits du résumé.*

$$\begin{array}{ll} PrI(z) & = \langle e_1, e_2, \ldots, e_k \rangle \\ avec & e_i \in z.A_i \text{ et } z.A_i \in \mathcal{F}(D^+_{A_i}), \quad 1 \leq i \leq k \end{array}$$

Le prototype idéalisé d'un résumé z est donc défini sur $\prod_{i=1}^{k}(z.A_i)$, qui donne un singleton sur chaque attribut du résumé z. Le sous-ensemble flou $z.A_i$ est représenté par un ensemble de triplets $(e_i, \mu_{e_i}, supp(e_i))$, où e_i est une étiquette linguistique sur l'attribut A_i, μ_{e_i} est le degré de satisfaction de e_i et $supp(e_i)$ est le support défini dans la présentation du système SaintEtiQ.

D'un point de vue sémantique, le *prototype idéalisé* offre à l'utilisateur une liste de descripteurs typiques d'un résumé dans une partition. Ce prototype *idéalisé* doit respecter la condition d'unicité, ainsi que l'objectif de décrire chaque attribut par un unique descripteur afin de simplifier l'interprétation de l'intention du résumé par l'utilisateur.

6.3.1.1 Construction d'un prototype *idéalisé*

La construction du prototype *idéalisé* est basée sur l'intention du résumé, c'est à dire qu'à partir des informations fournies dans la description intentionnelle d'un résumé notre objectif est de trouver la description prototypique de ce résumé au sein d'une partition. Le résultat de cette construction est un prototype qui decrit chacun des attributs du résumé par un seul descripteur, cette construction se fait donc attribut par attribut.

Pour cacluler ce prototype, nous nous sommes inspiré des travaux de construction des prototypes flous [61], sui se basent sur la typicité d'un élément dans une classe. Le calcul de la typicité utilise les mesures de comparaison comme la similarité intra-classe et la dissimilarité inter-classe. En effet, d'après B. Bouchon-Meunier et M. Rifqi [13] une mesure de comparaison entre deux sous-ensembles flous A et B prend en compte trois caractéristiques : l'intersection $A \cap B$ et la différence $A - B$ et $B - A$. Une mesure de comparaison peut évaluer la ressemblance de deux descriptions cette mesure est appelée mesure de *similitude* interne ou leurs différence et cette mesure est appelée mesure de *dissimilarité* externe.

Sur la base de ces mesures de ressemblance nous allons dans un premier lieu élaborer des mesures qui nous aideront à extraire le meilleur descripteur d'un résumé sur un attribut.

Calcul du meilleur descripteur

A notre sens, le meilleur descripteur est celui qui est typique au niveau de l'attribut d'un résumé. Cette typicité peut être simplement dûe au grand nombre de n-uplets décrit par ce descripteur et dans ce cas on parle de support d'un descripteur $supp(d)$. Mais vu que les descripteurs ayant les mêmes valeurs de support sont nombreux, la mesure de support est alors insuffisante pour trouver le meilleur descripteur. Pour ceci nous suivons une démarche pour extraire l'ensemble des meilleurs descripteurs au niveau du résumé puis au niveau de la partition.

Au sein du résumé interne (similitude). Afin de comparer les descripteurs sur un attribut au sein du même résumé, nous calculons un score pour chacun des descripteurs.

Pour un attribut $A \in \mathcal{A}$, l'intention $z.A$ du résumé z est un ensemble flou construit sur le domaine $\mathcal{F}_A^+$, l'ensemble des sous-ensembles flous de la liste des étiquettes linguistiques définies dans le BK correspondant à l'attribut considéré.

Calculer un score pour chacune des descriptions permet de mesurer sa qualité dans l'intention. Le score de chaque étiquette linguistique e (descripteur) sur un attribut A dans un résumé z est égal à la valeur maximale du support et du degré de satisfaction de l'étiquette :

$$Score_{z.A}(e) = \mu_{z.A}(e) \ \times \ supp_{z.A}(e)$$

avec, $supp(e) = \sum \omega(ct)$. Le support d'un descripteur d de l'intention d'un résumé noté $supp(e)$, représente le nombre d'instances effectivement décrites à l'aide de ce descripteur. Le poids $\omega(ct)$ d'un n-uplet candidat est égal à sa représentativité vis-à-vis des n-uplets d'origine.

Le calcul de ce score interne au résumé est insuffisant pour décrire un prototype unique dans la mesure où plusieurs descripteurs peuvent avoir les mêmes valeurs. Nous étendons la recherche du meilleur descripteur au niveau de la partition des résumés.

Pour une partition (dissimilarité). Afin de pouvoir choisir entre différents descripteurs à valeurs de score égales nous nous intéressons à leur pouvoir de discrimination.

Le pouvoir de discrimination d'un descripteur : nous considérons un ensemble de descripteur $\{e_1, \ldots, e_k\}$ sur un attribut A. Le descripteur discriminant sur $z, z \in P$ une partition de résumés, est celui qui se trouve dans z et pas dans les autres résumés de P. Plus formellement :

$$\begin{aligned} \forall z, z' \in P, \quad & e \in D_A^+ \\ & e \in supp(z.A) \ et \ e \in supp(z'.A)^1 \end{aligned}$$

e est le descripteur discriminant du résumé z sur la partition P.

Cette approche de calcul de prototype suivant l'intention du résumé permet de proposer un descripteur pour chaque attribut, et une représentation idéalisée du résumé. Cependant, et selon l'interprétation prototypique des concepts au sens de Eleanor Rosch [88], ce prototype a la particularité d'être éventuellement non présenté dans les données réelles décrites par ce résumé. Il est donc nécessaire de s'entourer de précautions lors du calcul du prototype, et de n'utiliser cette forme de représentation *idéalisée* qu'en première approximation. Cette méthode peut bien être utilisée dans le cas où l'utilisateur aimerait bien par exemple trouver la présentation prototypique d'un résumé tout en sachant que ce prototype ne correspond pas forcément à des enregistrements réels mais plutôt à ce que pourrait être des enregistrements typiques d'un ensemble de données résumées.

6.3.1.2 Application du prototype idéalisé

Nous avons appliqué cette méthode à une base de données commerciales d'une banque. Cette base contient 33733 n-uplets qui sont réecrits en 55546 n-uplets

candidats et génèrent 27 304 nœuds dont 14269 feuilles. Les dix attributs qui ont été décrits sont présentés dans la première colonne du tableau 6.6.

Attribut	Descripteur idéalisé	z feuille
Qte produit	very few	few
Nb carte bleue	none	none
Nb compte chèque	one	one
Segment comportemental	7	7
Flux créditeur	low	low
Nb d'opérations	average	average
Ressources totales	high	high
Nb retrait espèces	none	none
Durée relation	ancient	ancient
Age	old	old

Table 6.1 – Comparaison des deux prototypes du résumé racine R

Nous proposons dans l'algorithme 6.1, une méthode de recherche de l'ensemble des étiquettes linguistiques ou descripteurs les plus typiques sur l'intention d'un résumé.

Cet algorithme fait appel à la fonction *ExtractDescriptor()* présentée par l'algorithme 6.2, qui prend en entrée une liste de descripteurs D, sur un attribut donné A, un résumé z, et la partition en cours P. Elle fournit en sortie, un seul descripteur du résumé z sur l'attribut A. L'objectif de cette fonction est de garder le descripteur typique qui n'est pas représenté dans d'autres résumés de la partition sur l'attribut A.

Comme nous l'avons déjà indiqué, deux cas extrêmes se présentent : le premier est que la même liste de descripteurs extraits se trouve dans deux résumés (ou plus) différents de la partition manipulée, le second est qu'aucun des descripteurs conflictuels ne se trouve dans un autre résumé de sorte à ce qu'il ait un pouvoir de discrimination. Nous avons implémenter cet algorithme sur des données réelles.

Le tableau 6.2, montre une partie du résultat de l'extraction d'un descripteur prototypique à partir de la description intentionnelle du résumé. Dans ce tableau nous avons utilisé le résumé racine de la hiérarchie produite par SaintEtiQ appelé R. Le résultat final de cette extraction est présenté dans le tableau 6.3. Il s'agit du prototype idéalisé de la racine de la hiérarchie.

6.3.2 Prototype par extension d'un résumé

Dans cette section nous considérons l'extension d'un résumé z, notée R_z, afin de chercher le n-uplet candidat le plus typique. Le tableau 6.4 montre un exemple d'extension de résumé, défini comme l'ensemble des n-uplets candidats ct qui ont participé à la construction d'un résumé. Dans cette seconde méthode nous considérons le n-uplet avec tous les attributs sur lesquels il est défini, et

ALG. 6.1 – Recherche des descripteurs typiques d'un résumé

Entrée: z : un résumé, P : une partition.
Sortie: une liste de descripteurs.

```
DEBUT
D : une liste de descripteurs.
Score(d) = 0 : rel
pour tout attribut A ∈ 𝒜 faire
  répéter
    D ←<>.
    pour tout descripteur d ∈ z.A faire
      score(d) = μ(d).supp(d)
      /* Calculer le score de chaque descripteur d.*/
      si score(d) est maximal alors
        D.ajouter(d).
        /* Ajouter à la liste les descripteurs ayant la valeur maximale du
        score.*/
      fin si
    fin pour
    si |D| = 1 alors
      retourner(D).
    sinon
      ExtractDescriptor(D, P, A, z).
    fin si
  jusqu'à |D| = 1
fin pour
FIN .
```

ALG. 6.2 – Fonction d'extraction du descripteur typique

Entrée: D : une liste de descripteurs, P : partition, A : attribut, z : résumé.
Sortie: D : une liste avec un descripteur.

```
D' ←<> : une liste vide.
répéter
  pour tout z' ∈ P, avec z' ≠ z faire
    D' ←^{0+} z'.A.
    D ← D − (D ∩ D').
  fin pour
jusqu'à |D| = 1
retourner(D) .
```

Attribut	**Descripteurs**	**Descripteur Prototypique**
Qte produit	average(1, 38.5) few(1, 84.5) plenty(1, 3) very few(1, 98)	few
Nb carte bleu	none(1, 187) one(1, 37)	none
Nb compte chèque	many(1, 23) none(1, 8) one(1, 193)	one
Segment comportemental	1(1, 28) 2(1, 21) 3(1, 4) 4(1, 4) 5(1, 11) 6(1, 57) 7(1, 99)	7
Flux créditeur	average(1, 34.5) high(1, 11.5) low(1, 114) null(1, 64)	low
.....	..	

Table 6.2 – Résultat du prototype par intention du résumé racine R

few / Qte produit
None / Nb carte bleu
One / Nb compte chèque
7 / segment comportemental
Low / flux créditeur
Plenty / Nb d'opérations
High /ressources totales
None / Nb retraits espèce
Ancient / Duree relation
Old / Age

Table 6.3 – Prototype idéalisé du résumé racine R

nous comparons ces n-uplets entre eux pour garder à la fin le plus tyipque.

Considérer l'extension d'un résumé z revient à considérer toutes les feuilles du sous-arbre de ce résumé, vu qu'un résumé feuille est représenté par un seul descripteur sur chaque attribut. Prenons l'exemple de la figure 5.1 déjà présentée dans les deux chapitres précédents. Si on cherche à extraire le n-uplet candidat le plus représentatif de l'extension du résumé z_1, cela revient à trouver la meilleure feuille de la partition $\{z_{11}, z_{121}, z_{122}, z_{13}\}$.

T-candidat	A	B	C
ct_1	$ct_1.A$	$ct_1.B$	$ct_1.C$
ct_2	$ct_2.A$	$ct_2.B$	$ct_2.C$
ct_3	$ct_3.A$	$ct_3.B$	$ct_3.C$

Table 6.4 – Exemple d'extension d'un résumé sur trois attributs A, B et C

Nous proposons de calculer un degré de satisfaction, noté $sat(z)$, qui caractérise le sous-ensemble flou associé à chacune des feuilles. Nous avons besoin d'une fonction pour montrer le degré de satisfiabilité pour caractériser un résumé dans l'ensemble des feuilles prises en considération.

$$sat(z) = |R_z| \times min_A(supp(z.A))$$

– $|R_z|$ est le cardinal de l'extension du résumé z, le nombre de n-uplets candidats ct qu'il contient.
– $supp(z.A)$ est le support du descripteur $z.A$.

Le nombre de descripteurs k est égal au nombre d'attribut, $z.A_i$ représente le descripteur (un seul) qui décrit A_i dans z.

Definition 6.2 (Prototype par extension d'un résumé.)**.** *Soit $z_\star$ l'ensemble des feuilles du résumé z. Le prototype par extension d'un résumé z est la feuille la plus représentative de ce résumé, on le note :*

$$\begin{array}{ll} Pr(R_z) = & z'/sat_{z'} = max(sat_{z'_i}), \quad 1 \leq i \leq k \\ avec & z' \in z_\star,\ z_\star = \{z'_1 \dots\ z'_k\} \\ et & R_z = \bigcup_{z' \in z_\star} R_{z'} \end{array}$$

Selon cette définition, la satisfiabilité est maximale pour une feuille quand cette feuille contient le plus grand nombre de n-uplet candidat $max(|ct|)$, elle est pondérée par les degrés de satisfaction et par le support des n-uplets candidats au sein du résumé feuille.

6.3.2.1 Construction du prototype par extension

Sur les même données présentées dans la section 6.3.2, nous avons construit un prototype par extension tel qu'il a été défini. Nous proposons de montrer

le résultat de cette expérimentation sur le résumé racine. L'intérêt de choisir la racine est que son extension prend en considération toutes les feuilles de la hiérarchie. Le résultat obtenu est le résumé feuille *(z = R.0.1.1.1.1.0.0.1.1.0.0)* avec un degré $sat(z)$ le plus élevé et dont la description est proposée dans la troisième colonne du tableau 6.6.

Attribut	**Descripteur de z feuille**
Qte produit	few
Nb carte bleue	none
Nb compte chèque	one
Segment comportemental	7
Flux créditeur	low
Nb d'opérations	average
Ressources totales	high
Nb retrait espèces	none
Durée relation	ancient
Age	old

Table 6.5 – Résultat du prototype par extension du résumé racine R

Il est à noter que pour cette expérimentation, nous avons travaillé sur l'extension du résumé qui contient les n-uplets candidats ct et non pas les n-uplets originaux t.

Cette seconde méthode est efficace du point de vue de la qualité du prototype qu'elle fourni puisqu'il s'agit d'une présentation réelle et existante dans les données résumées contrairement à un résultat issu de la méthode idéalisée qui peut présenter des enregistrements inexistants.

En revanche, sur l'ensemble des feuilles d'un résumé, la possibilité d'avoir un degré de $sat(z)$ égal pour un ensemble de résumés feuilles n'est pas nulle. Sur la hiérarchie utilisée dans les expérimentations nous n'avons pas eu le cas d'avoir plusieurs feuilles avec la même valeur maximale $max(sat(z))$. Dans la section suivante nous proposons une autre méthode de construction de prototype flou en prenons en considération la faiblesse de la méthode de construction de prototype par extension du résumé.

6.3.3 Prototype combiné

Nous introduisons ici, la troisième méthode de calcul d'un prototype flou d'un résumé z. Nous proposons dans cette méthode de trouver parmi les résumés feuilles correspondants au résumé z, celui qui est le plus proche du prototype idéalisé. Pour ceci nous définissons une mesure de distance entre les descripteurs

de chaque attribut, cette distance est définie comme suit :

$$D(PrI(z), z') = \sum_{k}^{i=1} |z.A_i \cup z'.A_i|$$

où z' est une feuille de z, les descripteurs $z.A_i$ et $z'.A_i$ représentent l'attribut A_i sur z et z' respectivement, la différence est égale à 0 quand il s'agit du même descripteur et à 1 sinon. Cette mesure nous permet d'évaluer à quel point une feuille est proche du prototype idéalisé. On choisi parmi les résumés feuilles $z^\star$ de z, le résumé $z' \in z^\star$ qui a la valeur minimale de $D(PrI(z), z')$, puisque nous supposons que plus la distance diminue plus la similarité entre un résumé z' feuille de z et le prototype idéalisé de z augmente. Le prototype idéalisé est considéré ici comme un résumé feuille dont un attribut est décrit par un seul descripteur. Nous définissons alors ci-dessous le prototype final d'un résumé.

Definition 6.3 (Prototype combiné d'un résumé)**.** *Le prototype combiné d'un résumé z est l'une de ses feuilles qui a le plus grand degré de similarité avec le prototype idéalisé de z. Il est défini comme suit :*

$$\exists z' \in z_\star \text{ où } Pr(z) = z' \text{ tel que } D(PrI(z), z') \text{ est minimale}$$

où $PrI(z)$ est le prototype idéalisé de z, $z_\star$ est l'ensemble des feuilles de z.

6.3.3.1 Construction du prototype combiné

Nous reprenons l'exemple du résumé racine des données pris en considération dans la section 6.3.2, l'objectif est de calculer parmi les résumés feuilles celui qui la valeur minimale de la distance $D(PrI(R), R)$. Le prototype idéalisé de la racine est celui qui se trouve dans le tableau 6.3.

L'application de cette mesure nous fournit en résultat le résumé feuille *(z = R.0.1.1.1.1.0.0.1.1.0.0)*, le même que celui produit par la construction du prototype par extension. D'ailleurs nous remarquons sur le tableau 6.6 que les prototypes sont presque identiques, ce résultat a été vérifié pour la totalité des résumés de la hiérarchie, ceci s'explique par le fait que dans le jeu de données que nous utilisons la presque totalité des extensions des résumés feuilles contiennent un seul n-uplet.

Le tableau 6.6, montre une comparaison des résultats de l'application des trois méthodes proposées à une hiérarchie de données réelles, les descripteurs sont tous les mêmes sauf un qui décrit l'attribut "Quantité produit". Les trois méthodes utilisées ont chacune ses avantages et ses inconvénients :

– La première méthode du prototype idéalisé fournit un prototype de qualité qui peut dessiner une tendance permettant de donner une idée sur l'enregistrement qui peut être considéré comme typique pour représenter un résumé. Cependant cette méthode peut induire l'utilisateur en erreur du moment qu'elle peut fournir un prototype qui n'est pas cohérent avec les données existantes dans la base.

Attribut	prototype idéalisé	prototype sur extension	combiné
Qte produit	very few	few	few
Nb carte bleue	none	none	none
Nb compte chèque	one	one	one
Segment comportemental	7	7	7
Flux créditeur	low	low	low
Nb d'opérations	average	average	average
Ressources totales	high	high	high
Nb retrait espèces	none	none	none
Durée relation	ancient	ancient	ancient
Age	old	old	old

Table 6.6 – Comparaison des prototypes du résumé racine R

– La deuxième méthode du prototype par extension produit un résultat qui représente bien des données réelles, elles sont représentatives du résumé. Le point faible de cette méthode c'est que plusieurs résumés feuilles candidats au prototype peuvent prétendre représenter ce résumé d'une façon prototypique mais ne respecte pas la condition d'unicité dans ce cas.
– La troisième méthode vient combiner les deux autres. D'abord elle fait une première approximation en construisant un prototype idéalisé, ensuite elle cherche à trouver la feuille (parmi les résumés feuilles du résumé à représenter) qui se rapproche le plus de la version idéalisée, cette méthode est donc la plus complète.

Dans ce chapitre, nous avons introduit trois façons différentes pour construire des prototypes flous afin de représenter des résumés de bases de données volumineuses. Les trois méthodes proposées on été appliquées à une hiérarchie de résumés produite par le système SaintEtiQ. Elles consistent à calculer pour chaque résumé une représentation linguistique dite typique d'un résumé. Nous pensons que chacune des méthodes peut répondre à un besoin de l'utilisateur, mais nous considérons que la méthode hybride est la plus convaincante. Il s'agit dans cette méthode de sélectionner le résumé feuille qui est le plus proche du prototype idéalisé du résumé courant.

Conclusion

Dans cette partie nous nous sommes intéressés à la hiérarchie produite par SaintEtiQ, dans le but d'offrir une suite de fonctionnalités orientées utilisateur sous forme d'un outil d'analyse en ligne, qui permet d'explorer, de naviguer et de manipuler les résumés de la hiérarchie.

Pour ceci, nous avons présenté une structure multidimensionnelle appelée *partition de résumés* qui permet de modéliser les résumés linguistiques, et fournit à l'utilisateur un support pour la définition d'une algèbre de manipulation des partitions et des résumés avec un ensemble d'opérations, inspirées des opérateurs OLAP. Nous avons étudié cette algèbre, opérateur par opérateur, et nous avons présenté l'interface graphique d'exploration interactive, qui illustre la finalité du modèle et l'application de quelques opérateurs. Le modèle des partitions des résumés que nous proposons constitue le cœur de notre proposition pour faire évoluer le système SaintEtiQ vers une architecture décisionnelle basée sur les résumés flous. Nous pensons que ce modèle a été indispensable pour le module de présentation et d'analyse en ligne des résumés générés par SaintEtiQ.

Dans le deuxième chapitre de cette partie nous nous sommes focalisés sur la définition d'une algèbre de manipulation de ces partitions de résumés. En respectant une architecture décisionnelle type, nous avons détaillé l'adaptation des différents opérateurs d'analyse OLAP à notre modèle de partitions de résumés. Dans notre algèbre nous avons conservé la totalité des principes de base de SaintEtiQ, puisque nous agissons sur les résumés une fois générés par le système. Cette algèbre a repris les trois catégories d'opérateurs de manipulation d'analyse en ligne. Nous avons adapté les opérateurs de l'algèbre relationnelle, les opérateurs de granularité et les opérateurs de restructuration ayant une sémantique dans le modèle proposé.

Nous avons constaté que les résumés, présentés dans des partitions correspondant à des coupes de la hiérarchie, contiennent une grande quantité d'information à exploiter. Dans le troisième chapitre de cette partie, nous avons étudié comment représenter les résumés par leur forme typique en construisant des prototypes. Les notions de protoform et de prototype flou on été étudiées, pour les appliquer aux résumés de SaintEtiQ. La proposition de trois méthodes a été retenue et validée par des expériences sur des données réelles. Parmi ces trois méthodes, nous avons constaté que la plus complète est celle qui consiste à représenter un résumé par l'un de ses résumés feuilles, qui représente les éléments les plus typiques se rapprochant d'un prototype dit idéal, lui même calculé pour le résumé à représenter.

Chapitre 7

Conclusion générale

Bilan

Dans cette thèse nous avons abordé la thématique de l'analyse en ligne de résumés de bases de données. Ces résumés sont produits par le système SaintEtiQ proposé par G. Raschia et N. Mouaddib dans [84]. La thématique de l'analyse en ligne de données a pour sujet les grandes masses de données et pour objet de répondre aux interrogations des analystes et des décideurs. Elle s'intègre généralement dans les systèmes d'information décisionnels où elle est présentée par les manipulations proposées par les moteurs OLAP.

La première partie de ce document a été l'occasion de détailler les différentes méthodes qui se sont intéressées aux données traitées sous leur forme résumée ou agrégée. Nous avons tout d'abord étudié les systèmes décisionnels qui sont considérés comme les plus répandus pour le traitement de grandes masses de données à des fins analytiques. Notre intérêt s'est porté sur la grande capacité des opérateurs algébriques pour la manipulation des cubes de données traditionnels dont le but est de naviguer et d'explorer dans ces cubes. Nous avons souligné les avantages de ces systèmes OLAP, et notamment leur capacité exploratoire et analytique des données agrégées. Ensuite, nous avons fait un tour d'horizon des autres approches de compression sémantique des données. Parmi ces méthodes nous avons présenté plus en détail le système SaintEtiQ, plate-forme de résumé de grands volumes de données relationnelles. Ainsi, nous avons pu positionner le système SaintEtiQ et faire un rapprochement entre SaintEtiQ et un système décisionnel.

Dans la deuxième partie de ce document, nous nous sommes inspirés de la méthodologie d'analyse en ligne utilisée dans les systèmes décisionnels afin de l'adapter aux résumés linguistiques flous du système SaintEtiQ. Ces résumés de données sont représentés par une collection d'étiquettes linguistiques et chaque résumé fournit une représentation concise, par le biais d'un sous-ensemble flou de descripteurs sur chaque attribut, d'un ensemble de n-uplets de la base de données résumée. Nous avons constaté que la hiérarchie fournie par SaintEtiQ contient un grand nombre de résumés. Partis de ce constat, nous avons trouvé intéressant d'avoir une structure qui facilitera la navigation, l'exploration et la visualisation des résumés générés par SaintEtiQ. D'où l'idée de proposer un modèle multidimensionnel d'analyse en ligne des résumés linguistiques de données.

Cette proposition consiste à définir un cadre général pour l'exploration de

résumés de bases de données de taille significative.

En premier lieu nous avons défini un modèle de données logique appelé *partition de résumés*, par analogie avec les cubes de données OLAP, dans le but d'offrir à l'utilisateur final un outil de présentation des données sous forme condensée et adaptée à l'analyse. Dans l'espace des partitions de résumés, les relations entre les différentes partitions sont établies sur base du procédé de construction de résumés multi-niveaux développé dans SaintEtiQ.

Nous avons défini une collection d'opérateurs algébriques sur l'espace multidimensionnel de partitions de résumés [75, 76]. Ces opérateurs sont à la base d'une algèbre de manipulation des résumés. Cette algèbre prend en compte les spécificités du modèle de résumé que nous traitons et permet d'explorer succinctement le contenu de la base de données sans y accéder. Nous avons adapté la majorité des opérateurs d'analyse proposés dans les systèmes OLAP. Ainsi, nous avons identifié pour cette algèbre trois catégories d'opérateurs : les opérateurs de base (restriction, projection, union, ...) issus de l'algèbre relationnelle, les opérateurs de changement de granularité (*roll-up* et *le drill-down*) et les opérateurs de restructuration (rotation, permutation, ...). Ces résultats offrent de nouvelles perspectives pour l'exploitation effective des résumés dans un système décisionnel.

Pour compléter ce travail, nous nous sommes intéressés à la représentation des résumés et des partitions de résumés produits par SaintEtiQ, notamment pour en fournir une présentation claire et concise à l'utilisateur final. En effet, ces structures de données complexes, bien que définies à partir d'un langage contrôlé avec lequel l'utilisateur est familier, sont encore trop riches pour être interprétées correctement par des non spécialistes. Nous avons donc introduit une approche fondée sur les prototypes flous pour représenter des résumés de bases de données volumineuses. Appliquée à une hiérarchie de résumés produite par le système SaintEtiQ, l'approche tente d'extraire des prototypes selon trois orientations distinctes. La première proposition consiste à calculer pour chaque résumé une représentation linguistique dite « idéale » en identifiant le meilleur descripteur du résumé sur chacun de ses attributs. La seconde proposition vise à fournir comme prototype le résumé le plus précis situé dans le sous-arbre du résumé courant et qui traduit le plus fidèlement la représentation calculée précédemment. La troisième méthode consiste à trouver la feuille qui soit la plus proche dans sa description intentionnelle du prototype idéalisé. Le résultat de cette approche a été publié dans [74].

Cette thèse a été pour nous l'occasion d'étudier et d'adapter les techniques d'analyse en ligne de données et celles de construction de prototypes flous dans le cadre des résumés linguistiques. L'utilisation des techniques floues dans le système SaintEtiQ pour résumer des données permet de présenter les résumés selon des caractérisations linguistiques proches du langage naturel, ce qui permet une certaine flexibilité dans l'exploration de la hiérarchie, ainsi qu'une nuance dans les jugements et les décisions. Dans le dernier chapitre concernant la construction de prototypes flous pour représenter des résumés, nous avons adopté une

démarche qui tire profit des travaux réalisés en sciences cognitives sur le processus humain de catégorisation. Ainsi les prototypes calculés tendent à reproduire les mécanismes de synthèse en fonction chez l'homme.

Evolutions et perspectives

En guise d'extension de nos travaux de recherche, nous avons envisagé à différentes possibilités à explorer :

- La première évolution de nos travaux concerne l'aspect applicatif de la proposition détaillée dans cette thèse. En effet, ce travail a été proposé dans le cadre de manipulation de grands volumes de données. Il manque donc la validation du modèle et de l'algèbre proposée dans ce travail sur des bases de données réelles et massives.
- La deuxième perspective serait d'appliquer notre modèle aux résumés multimédias. Le terme multimédia désigne des données aussi diverses que des images, des sons ou des vidéos. Ces dernières années, un certain nombre de systèmes de recherche d'information multimédia et de systèmes de gestion de bases de données multimédia ont été développés afin d'aider l'usager à retrouver des documents correspondant à un besoin spécifique, parmi de très grandes collections. Pour ceci SaintEtiQ a déjà été utilisé sur des bases d'images dans [92, 64, 91]. Il apparaît donc intéressant de valider notre modèle sur des résumés d'images.
- Par ailleurs, le travail présenté dans le chapitre 6 montre que l'utilisation des protoforms pour la représentation des résumés, mérite d'être étendue. Il serait notamment très intéressant de travailler sur les possibilités de caractériser les résumés en apportant à l'utilisateur une information complémentaire à celle des prototypes flous. Cette caractérisation pourrait permettre d'évaluer un résumé en calculant des cœfficients à partir de ses propriétés afin de l'identifier par exemple comme majoritaire, discriminant ou exceptionnel.

Bibliographie

[1] A. Abello, J. Samos et F. Saltor. Implementing operations to navigate semantic star schemas. In *DOLAP '03: Proceedings of the 6th ACM international workshop on Data warehousing and OLAP*, pages 56–62, New York, NY, USA, 2003. ACM Press.

[2] R. Agrawal, A. Gupta et S. Sarawagi. Modeling multidimensional databases. *in Proc. of ICDE*, pages 232–243, *April* 1997.

[3] R. Agrawal, T. Imielinski et A. N. Swami. Mining association rules between sets of items in large databases. In Peter Buneman et Sushil Jajodia, réds., *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, pages 207–216, Washington, D.C., 1993.

[4] R. Agrawal et R. Srikant. A possibilistic approach to clusterin. In J.M. Krishnapuram, R.; Keller, réd., *Fuzzy Systems, IEEE Transactions on Volume 1*, pages 98 – 110. Morgan Kaufmann, 1993.

[5] R. Agrawal et R. Srikant. Fast algorithms for mining association rules. In Jorge B. Bocca, Matthias Jarke et Carlo Zaniolo, réds., *Proc. 20th Int. Conf. Very Large Data Bases, VLDB*, pages 487–499. Morgan Kaufmann, 1994.

[6] S. Babu, G. Minos et R. Rajeev. SPARTAN: A model-based semantic compression system for massive data tables. In *Proc. of the 2001 ACM Intl. Conf. on Management of Data (SIGMOD 2001)*, pages 283–295, May 2001.

[7] E. Baralis, S. Paraboschi et E. Teniente. Materialized views selection in a multidimensional database. In *The VLDB Journal*, pages 156–165, 1997.

[8] J. C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Kluwer Academic Publishers, Norwell, MA, USA, 1981.

[9] I. Blanco, D. Sánchez, J. M. Serrano et M. A. V. Miranda. A new proposal of aggregation functions: The linguistic summary. In *IFSA*, pages 127–134, 2003.

[10] M. Blaschka, C. Sapia, G. Höflng et B. Dinter. Finding your way through multidimensional data models. In *DEXA '98: Proceedings of the 9th International Workshop on Database and Expert Systems Applications*, page 198, Washington, DC, USA, 1998. IEEE Computer Society.

[11] P. Bosc, D. Dubois et H. Prade. Fuzzy functional dependencies and redundancy elimination. *JASIS*, 49(3):217–235, 1998.

[12] P. Bosc, O. Pivert et L. Ughetto. On data summaries based on gradual rules. In *Fuzzy Days*, pages 512–521, 1999.

[13] B. Bouchon-Meunier, M. Rifqi et S. Bothorel. Towards general measures of comparison of objects. pages 143–153, 1996.

[14] J. F. Boulicaut, P. Marcel et C. Rigotti. Query driven knowledge discovery in multidimensional data. In *DOLAP*, pages 87–93, 1999.

[15] A. W. Bragg. Data manipulation languages for statistical databases - the statistical analysis system (SAS). In Harry K. T. Wong, réd., *Proceedings of the First LBL Workshop on Statistical Database Management, Melno Park, California, USA, December 2-4, 1981*, pages 147–150. Lawrence Berkeley Laboratory, 1981.

[16] L. Cabibbo et R. Torlone. Querying multidimensional databases. In *DBLP-6: Proceedings of the 6th International Workshop on Database Programming Languages*, pages 319–335, London, UK, 1998. Springer-Verlag.

[17] Y. W. Choong, D. Laurent et P. Marcel. Computing appropriate representations for multidimensional data. In *DOLAP*, 2001.

[18] Claude Chrisment, Geneviève Pujolle, Franck Ravat, Olivier Teste et Gilles Zurfluh. Les entrepôts de données. In Gilles Zurfluh, réd., *Traité Informatique des Techniques de l'Ingénieur - H3870*, page 10. Techniques de l'Ingénieur, février 2005.

[19] E. F. Codd. Providing OLAP (On-Line Analytical Processing) to user-analysts: An IT mandate. In *Technical Reports*. IBM, 1993.

[20] J. C. Cubero, J. M. Medina, O. Pons et M. A. V. Miranda. Data summarization in relational databases through fuzzy dependencies. *Inf. Sci.*, 121(3-4):233–270, 1999.

[21] J. C. Cubero et M. A. Vila. A new definition of fuzzy functional dependency in fuzzy relational databases. *International Journal of Intelligent Systems*, 9(5):pp:441–448, 1994.

[22] A. Datta et H. Thomas. The cube data model: a conceptual model and algebra for on-line analytical processing in data warehouses, 1999.

[23] M. Delgado, C. Molina, D. Sánchez, L. R. Ariza et M. A. V. Miranda. A flexible approach to the multidimensional model: The fuzzy datacube. In *CAEPIA*, pages 26–36, 2003.

[24] E. Diday. Une nouvelle méthode en classification automatique et reconnaissance des formes : la méthode des nuées dynamiques. *In Rev. Statist. Appl*, 19:19–33, 1971.

[25] E. Forgy. Cluster analysis of multivariate data: Efficiency vs. interpretability of classifications. *Biometrics, 21(3):768*, 1965.

[26] J. M. Franco. *Le Data Warehouse*. Eyrelles, 1997.

[27] O. François et P. Leray. Etude comparative d'algorithmes d'apprentissage de structure dans les réseaux bayésiens. *Journal électronique d'intelligence artificielle*, 5(39):1–19, 2004.

[28] H. Frigui et O. Nasraoui. Unsupervised learning of prototypes and attribute weights. *Pattern Recognition*, 37(3):567–581, 2004.

[29] S. P. Ghosh. Panel discussion - statistical relational model. In *SSDBM*, pages 373–387, 1988.

[30] A. Giacometti, D. Laurent, P. Marcel et H. Mouloudi. A new way of optimizing OLAP queries. In *In Actes des 20ièmes journées Bases de Données Avancées.*, pages 109–128, Lyon, France, October 2004.

[31] J. F. Goglin. La construction du data warehouse, du data mart au dataweb. *Nouvelles Technologies Informatiques*, HERMES, 2001.

[32] J. Gray, S. Chaudhuri, A. Bosworth, A. Layman, D. Reichart, M. Venkatrao, F. Pellow et H. Pirahesh. Data cube: A relational aggregation operator generalizing group-by, cross-tab, and sub-totals. *Data Min. Knowl. Discov.*, 1(1):29–53, 1997.

[33] A. Gupta, I. S. Mumick et K. A. Ross. Adapting materialized views after redefinitions. In Michael J. Carey et Donovan A. Schneider, réds., *Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data, San Jose, California, May 22-25, 1995*, pages 211–222. ACM Press, 1995.

[34] M. Gyssens et L. V. S. Lakshmanan. A foundation for multi-dimensional databases. In *VLDB '97: Proceedings of the 23rd International Conference on Very Large Data Bases*, pages 106–115, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc.

[35] M. S. Hacid, P. Marcel et C. Rigotti. A rule-based CQL for 2-dimensional tables. In *CDB*, pages 92–104, 1997.

[36] J. Han, Y. Cai et N. Cercone. Knowledge discovery in databases: An attribute-oriented approach. In Li-Yan Yuan, réd., *Proceedings of the 18th International Conference on Very Large Databases*, pages 547–559, San Francisco, U.S.A., 1992. Morgan Kaufmann Publishers.

[37] J. Han, Y. Fu, Y. Huang, Y. Cai et N. Cercone. DBLearn: a system prototype for knowledge discovery in relational databases. In *SIGMOD '94: Proceedings of the 1994 ACM SIGMOD international conference on Management of data*, page 516, New York, NY, USA, 1994. ACM Press.

[38] J. Han, Y. Fu, W. Wang, J. Chiang, W. Gong, K. Koperski, D. Li, Y. Lu, A. Rajan, N. Stefanovic, B. Xia et O. R. Zaiane. DBMiner: A system for mining knowledge in large relational databases. In *Proc. 1996 Int'l Conf. on Data Mining and Knowledge Discovery (KDD'96)*, pages 250–255, Portland, Oregon, 1996.

[39] J. Han, J. Wang, G. Dong, J. Pei et K. Wang. CubeExplorer: online exploration of data cubes. In *SIGMOD02: Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data*, pages 626–626, New York, NY, USA, 2002. ACM Press.

[40] V. Harinarayan, A. Rajaraman et J. D. Ullman. Implementing data cubes efficiently. In *SIGMOD '96: Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*, pages 205–216, New York, NY, USA, 1996. ACM Press.

[41] D. A. Huffman. A method for the construction of minimum-redundancy codes. *PIRE*, 40(9):1098–1101, September 952.

[42] W. H. Inmon. *Building the Data Warehouse*. John Wiley & Sons, Inc., New York, NY, USA, 1996.

[43] H. V. Jagadish, L. V. S. Lakshmanan et D. Srivastava. What can hierarchies do for data warehouses? In Malcolm P. Atkinson, Maria E. Orlowska, Patrick Valduriez, Stanley B. Zdonik et Michael L. Brodie, réds., *VLDB'99, Proceedings of 25th International Conference on Very Large Data Bases, September 7-10, 1999, Edinburgh, Scotland, UK*, pages 530–541. Morgan Kaufmann, 1999.

[44] H. V. Jagadish, J. Madar et R. T. Ng. Semantic compression and pattern extraction with fascicles. In *Proc. of 25th Intl. Conf. on Very Large Data Bases (VLDB99)*, pages 186–198, 1999.

[45] H. V. Jagadish, R. T. Ng, B. C. Ooi et A. K. H. Tung. Itcompress: An iterative semantic compression algorithm. In *20th Intl. Conf. on Data Engineering*, page 646, 2004.

[46] J. Kacprzyk et S. Zadronzny. On interactive linguistic summarization of databases via a fuzzy-logic-based querying add-on to microsoft access. In *Fuzzy Days*, pages 462–472, 1999.

[47] J. Kacprzyk et S. Zadrozny. Protoforms of linguistic data summaries: Towards more general natural-language-based data mining tools. In *HIS*, pages 417–425, 2002.

[48] J. Kacprzyk et S. Zadrozny. Linguistic database summaries and their protoforms: towards natural language based knowledge discovery tools. *Inf. Sci.*, 173(4):281–304, 2005.

[49] A. Kaufmann. *Introduction à la théorie des sous-ensembles flous : Applications à la classification et à la reconnaissance des formes, aux automates et aux systèmes, aux choix des critères.*, volume 3. Masson, 1975.

[50] F. Khalfallah et K. Mellouli, réds. *Apprentissage de la structure d'un réseau bayésien à partir d'une base de données*, volume 18(2) pp.195-228 of *RSTI, Revue d'intelligence artificielle, Serie RIA*. Hermès, 2004.

[51] R. Kimball. *The data warehouse toolkit: practical techniques for building dimensional data warehouses*. John Wiley & Sons, Inc., New York, NY, USA, 1996.

[52] L. V. S. Lakshmanan, J. Pei et J. Han. Quotient cube: How to summarize the semantics of a data cube. In *VLDB*, pages 778–789, 2002.

[53] L. V. S. Lakshmanan, J. Pei et Y. Zhao. QC-trees: an efficient summary structure for semantic OLAP. In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, pages 64–75. ACM Press, 2003.

[54] L. V. S. Lakshmanan, J. Pei et Y. Zhao. SOCQET: semantic OLAP with compressed cube and summarization. In *SIGMOD '03: Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, pages 658–658, New York, NY, USA, 2003. ACM Press.

[55] A. Laurent. *Bases de données multidimensionnelles floues et leur utilisation pour la fouille de données*. Thèse de Doctorat, Université de Paris6, dec 2002.

[56] A. Laurent. Querying fuzzy multidimensional databases: unary operators and their properties. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, 11(Supplement):31–45, 2003.

[57] O. Lebeltel, P. Bessière, J. Diard et E. Mazer. Programmation bayésienne des robots. *Revue d'Intelligence Artificielle*, 18(2):261–298, 2004.

[58] H. J. Lenz et A. Shoshani. Summarizability in OLAP and statistical data bases. In *SSDBM '97: Proc. of the 9th Intl. Conf. on Scientific and Statistical Database Management*, pages 132–143. IEEE Computer Society, 1997.

[59] P. Leray et O. Francois. Réseaux bayésiens pour la classification – méthodologie et illustration dans le cadre du diagnostic médical. *Revue d'Intelligence Artificielle*, 18/2004:169–193, 2004.

[60] M.-J. Lesot. Similarity, typicality and fuzzy prototypes for numerical data. In *In 6th European Congress on Systems Science, Workshop "Similarity and resemblance"*, 2005.

[61] M.-J. Lesot, L. Mouillet et B. Bouchon-Meunier. Fuzzy prototypes based on typicality degrees. In *Fuzzy Days04*, 2004.

[62] M. Levene et G. Loizou. Why is the snowflake schema a good data warehouse design? *Inf. Syst.*, 28(3):225–240, 2003.

[63] C. Li et X. S. Wang. A data model for supporting on-line analytical processing. In *CIKM*, pages 81–88, 1996.

[64] E. Loisant, R. Saint-Paul, J. Martinez, G. Raschia et N. Mouaddib. Browsing clusters of similar images. In *Actes des 19e Journées Bases de Donnéees Avancéees (BDA'2003)*, pages 109–128, Lyon, France, October 2003.

[65] J. Macqueen. Some methods for classification and analysis of multivariate observations. *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, Vol. 1 pp. 281-297, 1967.

[66] A. S. Maniatis, P. Vassiliadis, S. Skiadopoulos et Y. Vassiliou. Advanced visualization for olap. In *DOLAP '03: Proceedings of the 6th ACM international workshop on Data warehousing and OLAP*, pages 9–16, New York, NY, USA, 2003. ACM Press.

[67] P. Marcel. *Manipulation des données multidimensionnelles.* Thèse de Doctorat, Université de Lyon, dec 1998.

[68] P. Marcel. Modeling and querying multidimensional databases: An overview, 1999.

[69] R. Ben Messaoud, K. Aouiche et C. Favre. Une approche de construction d'espaces de représentation multidimensionnels dédiés à la visualisation. In *1ère journée sur les Entrepôts de Données et l'Analyse en Ligne (EDA 05), Lyon*, volume B-1 of *Revue des Nouvelles Technologies de l'Information*, pages 34–50, Toulouse, Juin 2005. Cépaduès Editions.

[70] R. Ben Messaoud, O. Boussaid et S. Loudcher Rabaséda. Using a factorial approach for efficient representation of relevant OLAP facts. In *Proc. of the Seventh International Baltic Conference on Databases and Information Systems (DB&IS 2006)*, July 2006.

[71] Josiane Mothe, Claude Chrisment et Joel Alaux. Visualisation globale de collections de documents sous forme d'hypercube - Le système DocCube . In *Journées francophones d'Extraction et de Gestion des Connaissances, EGC2002 , Montpellier, France, 21/01/02-23/01/02*, pages 131–142. Hermès, janvier 2002.

[72] Josiane Mothe, Claude Chrisment, Taoufiq Dkaki, Bernard Dousset et Said Karouach. Combining mining and visualization tools to discover the geographic structure of a domain. *Computers, Environment and Urban Systems, Geographic Information Retrieval*, Hors-série(4):460–484, juillet 2006.

[73] Josiane Mothe, Claude Chrisment, Bernard Dousset et Joel Alaux. DocCube: Multi-Dimensional Visualisation and Exploration of Large Document Sets . *Journal of the American Society for Information Science and Technology, JASIST, Special topic section: web retrieval and mining*, 7(54):650–659, mars 2003.

[74] L. Naoum. Représentation de résumés de base de données par prototypes flous. In *à paraître, 12th LFA Conference (LFA 2006)*, Toulouse, France, Octobre 2006.

[75] L. Naoum, G. Raschia et N. Maouddib. Towards on-line analytical processing for database summaries: The core algebra. In *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, volume 1, pages 3625–3632, Vancouver, Canada, July 2006.

[76] L. Naoum, G. Raschia et N. Mouaddib. Querying fuzzy summaries of databases: Unary operators and their properties. In *Proceedings of the Joint 4th EUSFLAT & 11th LFA Conference (EUSFLAT-LFA'2005)*, pages 1194–1200, Barcelona, Spain, September 2005.

[77] J. A. Olivas et F. P. Romero. FPKD: Fuzzy prototypical knowledge discovery. application to forest fire prediction. In *Proceedings of the SEKE'2000, Knowledge Systems Institute*, 2000.

[78] T. B. Pedersen et C. S. Jensen. Multidimensional data modeling for complex data. In *ICDE '99: Proceedings of the 15th International Conference on Data Engineering*, pages 336–346, Washington, DC, USA, 1999. IEEE Computer Society.

[79] T. B. Pedersen, C. S. Jensen et C. E. Dyreson. Supporting imprecision in multidimensional databases using granularities. In *Statistical and Scientific Database Management*, pages 90–101, 1999.

[80] E. Pourabbas et M. Rafanelli. Hierarchies and relative operators in the OLAP environment. *SIGMOD Rec.*, 29(1):32–37, 2000.

[81] H. Prade et C. Testemale. Generalizing database relational algebra for the treatment of incomplete/uncertain information and vague queries. *Inf. Sci.*, 34(2):115–143, 1984.

[82] K. V. S. V. N. Raju et A. K. Majumdar. Fuzzy functional dependencies and lossless join decomposition of fuzzy relational database systems. *ACM Trans. Database Syst.*, 13(2):129–166, 1988.

[83] G. Raschia. *SAINTETIQ: une approche floue pour la génération de résumés à partir de bases de données relationnelles.* Thèse de Doctorat, Université de Nantes.

[84] G. Raschia et N. Mouaddib. A fuzzy set-based approach to database summarization. *Int. Journal of Fuzzy Sets and Systems*, 129(2):137–162, July 2002.

[85] D. Rasmussen et R. R. Yager. Summary SQL - a Fuzzy Tool For Data Mining. *Intell. Data Anal.*, 1(1-4):49–58, 1997.

[86] F. Ravat, O. Teste et G. Zurfluh. Manipulation et fusion de données multidimensionnelles. In RNTI-E-3 Revue des Nouvelles Technologies de l'Information, réd., *Extraction et Gestion des Connaissances (EGC'2005) Vol. I*, pages 349–354, Paris, France, Janvier 2005. Cépadues (ed.).

[87] M. Rifqi. Constructing prototypes from large databases. In *IPMUInformation Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 301–306, 1996.

[88] E. Rosch. Principles of categorization. In *In E. Rosch and B. Lloyd, editors, Cognition and categorization*, pages 27–48. Lawrence Erlbaum associates, 1978.

[89] E. H. Ruspini. A new approach to clustering. 15(1):22–32, juillet 1969.

[90] R. Saint-Paul. *Une architecture pour le résumé en ligne de données relationnelles et ses applications.* Thèse de Doctorat, Université de Nantes.

[91] R. Saint-Paul, G. Raschia et N. Maouddib. Prototyping and browsing image databases using linguistic summaries. In *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, volume 1, pages 476–481, Honolulu, Hawaii (USA), May 2002. IEEE Press.

[92] R. Saint-Paul, G. Raschia et N. Mouaddib. Image database summarization with the saintetiq system. In *Proc. of the 9th Int. Conf. on Information*

Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU'2002), pages 1179–1186, Annecy, France, July 2002.

[93] R. Saint-Paul, G. Raschia et N. Mouaddib. Résumé de bases de données : application au domaine bancaire. *Technique et Science Informatiques (TSI)*, 22:1353–1379, 2003.

[94] A. Shoshani. OLAP and statistical databases: similarities and differences. In *PODS '97: Proceedings of the sixteenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*, pages 185–196, New York, NY, USA, 1997. ACM Press.

[95] O. Teste. *Modélisation et manipulation d'entrepôts de données complexes et historisées*. Thèse de Doctorat, Univeristé Paul Sabatier de Toulouse.

[96] J. D. Ullman. Efficient implementation of data cubes via materialized views. In *KDD*, pages 386–388, 1996.

[97] A. A. Vaisman et A. O. Mendelzon. A temporal query language for olap: Implementation and a case study. In *DBPL '01: Revised Papers from the 8th International Workshop on Database Programming Languages*, pages 78–96, London, UK, 2002. Springer-Verlag.

[98] P. Vassiliadis. Modeling multidimensional databases, cubes and cube operations. In *SSDBM '98: Proceedings of the 10th International Conference on Scientific and Statistical Database Management*, pages 53–62, Washington, DC, USA, 1998. IEEE Computer Society.

[99] P. Vassiliadis et T. Sellis. A survey of logical models for olap databases. *SIGMOD Rec.*, 28(4):64–69, 1999.

[100] J. Widom. Research problems in data warehousing. In *CIKM '95, Proceedings of the 1995 International Conference on Information and Knowledge Management, November 28 - December 2, 1995, Baltimore, Maryland, USA*, pages 25–30. ACM, 1995.

[101] www.decisionnel.net.

[102] R. R. Yager. Aggregation operators and fuzzy systems modeling. *Fuzzy Sets Syst.*, 67(2):129–145, 1994.

[103] R. R. Yager et T. C. Rubinson. Linguistic summaries of databases. In *Proc. Decision and Control*, pages 1094–1097, San Diego, USA, 1981. IEEE Decision and Control.

[104] L. Zadeh.. A note on prototype set theory and fuzzy sets. In *Cognition 12*, pages 291–297, 1982.

[105] L. A. Zadeh. Fuzzy sets. *Information and Control*, 8(3):338–353, 1965.

[106] L. A. Zadeh. A prototype-centered approach to adding deduction capability to search engines – the concept of protoform. *BISC Seminar, 2002, University of California, Berkley*, 2002.

[107] S. Zadrozny et J. Kacprzyk. FQUERY for access: towards human consistent querying user interface. In *SAC '96: Proceedings of the 1996 ACM symposium on Applied Computing*, pages 532–536, New York, NY, USA, 1996. ACM Press.

[108] S. Zadrozny, J. Kacprzyk et M. Gola. Towards human friendly data mining: Linguistic data summaries and their protoforms. In *ICANN (2)*, pages 697–702, 2005.

[109] J. Ziv et A. Lempel. A universal algorithm for sequential data compression. *IEEE Transactions on Information Theory*, 23(3):337–343, 1977.

Liste des tableaux

Partie I — État de l'art

Partie II — Vers un processus d'analyse en ligne de résumés flous

Table des figures

Partie I — État de l'art

Partie II — Vers un processus d'analyse en ligne de résumés flous

Table des exemples

Table des matières

Partie II — Vers un processus d'analyse en ligne de résumés flous

Annexes

Annexe A

Glossaire & Notations.

Glossaire

– *BD, Base de données (DB, Database)*
Une base de données regroupe en un ensemble structuré les informations nécessaires à une ou plusieurs applications de l'entreprise. L'accès aux informations se fait grâce au SGBD (Système de Gestion de Base de Données). Le modèle de données relationnel, le plus courant aujourd'hui, mémorise les relations existant entre les informations dans des tables.

– *Base multidimensionnelle*
Pour pouvoir analyser les données représentant l'activité d'une entreprise, il faut pouvoir les modéliser suivant des axes. Ainsi, pour prendre l'exemple le plus courant, le chiffre d'affaires par catégorie de client sur un produit donné se décline sur trois axes au minimum : chiffre d'affaires, catégorie de clients, et produit. De nombreux autres axes peuvent être définis, notamment en fonction de la zone géographique, du prix, ou d'un commercial de l'équipe en charge des opérations. Une base de données multidimensionnelle stocke les données de manière à permettre ce type d'analyses.

– *Entrepôt de données (Data warehouse)*
Structure informatique dans laquelle est centralisé un volume important de données consolidées à partir des différentes sources de renseignements d'une entreprise (notamment les bases de données internes). L'organisation des données est conçue pour que les personnes intéressées aient accès rapidement et sous forme synthétique à l'information stratégique dont elles ont besoin pour la prise de décision.

– *Magasin de données (Data mart)*
Entrepôt de données départemental. Sous-ensemble d'un entrepôt de données, contenant des informations se rapportant à un secteur d'activité particulier de l'entreprise ou à un métier qui y est exercé (commercial, marketing, comptabilité, etc.).

– *Fouille de données (Data mining)*
Technique d'analyse utilisant un logiciel pour dénicher des tendances ou des corrélations cachées parmi des masses de données, ou encore pour détecter

des informations stratégiques ou découvrir de nouvelles connaissances, en s'appuyant sur des méthodes de traitement statistique.

- *ETL (Extraction, Transformation and Loading)*
  Outil informatique destiné à extraire des données de diverses sources (bases de données de production, fichiers, Internet, etc.), à les transformer et à les charger dans un entrepôt de données.

- *Informatique décisionnelle (BI, Business Intelligence)*
  Système interprétant des données complexes permettant aux dirigeants d'entreprise de prendre des décisions en connaissance de cause. Les données sont analysées selon plusieurs dimensions (type de produits, régions et saisons par exemple). De plus en plus, l'informatique décisionnelle se rapproche de l'intelligence d'affaires, où un système informatique permet la recherche active et l'exploitation, sur le plan décisionnel, de l'ensemble des renseignements stratégiques essentiels qu'une entreprise doit posséder, si elle veut faire face à la concurrence et occuper la première place, dans son secteur d'activités.

- *OLAP (On-Line Analytical Processing)*
  Technologie créée par E. F. Codd, père des bases de données relationnelles, et conçue pour répondre aux besoins d'analyse des applications de gestion. Les systèmes OLAP fournissent une vue multidimensionnelle des données agrégées issues d'un data warehouse.

- *DOLAP (Dynamic On-Line Analytical Processing)*
  Technologie qui permet la navigation dans les données de façon dynamique.

- *HOLAP (Hybrid On-Line Analytical Processing)*
  Technologie qui prend en compte à la fois les bases de données relationnelles et multidimensionnelles.

- *MOLAP (Multidimentional On Line Analytical Processing)*
  Technologie qui stocke physiquement les données dans une base multidimensionnelle.

- *ROLAP (Relational On Line Analytical Processing)*
  Technologie qui stocke physiquement les données dans un SGBD relationnel. Un cube multidimensionnel est fourni en réponse à chaque interrogation de l'utilisateur.

- *Reporting*
  Production de rapports pré conçus pour la diffusion d'informations analytiques et synthètiques au niveau de la prise de décision.

– *SIAD (Système informatisé d'aide à la décision), DSS (Decision support system)*
Système informatique intégré, conçu spécialement pour la prise de décision, et qui est destiné plus particulièrement aux dirigeants d'entreprise. Le système d'aide à la décision est un des éléments du système d'information de gestion. Il se distingue du système d'information pour dirigeants, dans la mesure où sa fonction première est de fournir non seulement l'information, mais aussi les outils d'analyse nécessaires à la prise de décision. Ainsi, il est habituellement constitué de programmes, d'une ou de plusieurs bases de données, internes ou externes, et d'une base de connaissances. Il fonctionne avec un langage et un programme de modélisation permettant aux dirigeants d'étudier différentes hypothèses en matière de planification et d'en évaluer les conséquences.

NOTATIONS

Voici les différentes notations utilisées dans le système SaintEtiQ.

- R : table relationnelle.
- $\mathcal{A}$: ensemble des attributs d'une relation.
- D_A : le domaine de base de l'attribut A.
- D_A^+ : le domaine réécrit de l'attribut A (ensemble de descripteurs).
- BK : Connaissances de domaine, i.e. une collection d'ensembles flous.
- $t = \langle t.A_1, ..., t.A_n \rangle$: n-uplet où $A_i \in \mathcal{A}$ et $t.A_i \in D_{A_i}$ est la valeur de t sur A_i.
- $ct = \langle ct.A_1, ..., ct.A_n \langle$: n-uplet candidat avec $ct.A_i \in \mathcal{F}(D_{A_i}^+)$ et $||ct.A_i|| = 1$.
- $z = \langle z.A_1, ..., z.A_n \langle$: n-uplet de la relation R, intention d'un résumé z avec $z.A_i \in \mathcal{F}(D_{A_i}^+)$.
- R_z : extension du résumé z en termes de n-uplets candidats $Rz = ct_1, ..., ct_n$.
- $card(R_z)$: cardinal relatif de l'extension de z. $card(Rz) = \sum_{ct \in R_z} w(ct)$.
- $w(ct)$: poids du candidat ct dans la représentation de t.

Annexe B

Propositions industrielles des solutions décisionnelles.

Cette annexe présente les principales offres commerciales du marché des solutions décisionnelles. Ces solutions sont classées par ordre alphabétique et présentées suivant leurs différentes fonctionnalités.

Brio[1]

Brio Software propose une solution complète de business intelligence dédiée à la requête et l'analyse des données, au reporting et à la diffusion d'informations, dont la vocation est d'aider les entreprises à optimiser leurs performances. La plate-forme décisionnelle de Brio se compose de trois modules :

- Brio Intelligence : suite intégrée composée d'outils d'analyse, de requête et de reporting analytique pour les environnements Web et Client/Serveur.
- Brio Report : serveur à hautes performances pour les rapports manipulant des volumes de données importants et couvrant tous les besoins de reporting d'entreprise.
- Brio Portal : portail décisionnel d'entreprise.

Les principaux avantages de cette solution résident dans la satisfaction de requêtes dynamiques, et l'utilisation de technologie OLAP permettant ainsi la diffusion d'informations de reporting analytique dans un environnement client/serveur ou web. En revanche son grand inconvénient réside dans l'absence d'outil analytique.

Business Object[2]

La solution décisionnelle de BO est une suite d'outils intégrés de business intelligence, elle est considérée comme numéro un sur le marché avec son software "Business Objects Analytics" qui représente une suite intégrée d'applications analytiques d'entreprise. Cette offre complète permet aux utilisateurs d'accéder,

[1] Brio Software, http://www.brio.com/fr/
[2] http://www.france.businessobjects.com/

d'analyser et de partager l'information dans l'entreprise et à l'extérieur de l'entreprise. Ces principales fonctionnalités sont :
– Portail et diffusion de l'information,
– interrogation, reporting et analyse,
– applications analytiques,
– déploiement d'entreprise,
– intégration de données.

Cognos[3]

Cognos est un outil décisionnel des plus utilisés dans les entreprises, il est constitué d'un ensemble d'outils d'analyse de données facilitant la prise de décision. Ce sont des applications intégrées au sein d'une offre complète de business intelligence, depuis l'ETL jusqu'au portail décisionnel. La plupart de ses fonctionnalités et domaines d'application sont :
– analyse des ventes et des achats,
– analyse comptable et financière,
– analyse des stocks et de la demande,
– analyse multidimensionnelle à l'échelle de l'entreprise : *PowerPlay*,
– analyse multidimensionnelle sur le Web : *PowerPlay Web*.

Crystal Decision[4]

Crystal Decision propose la solution décisionnelle appelée CRYSTAL ENTREPRISE, c'est une suite logicielle intégrée et complètement dédiée au Web pour le reporting, l'analyse et la diffusion d'informations, à l'échelle de l'entreprise. L'avantage majeur de cette solution est sa prise en charge de différentes plates-formes windows et unix. Sa principale fonctionnalité est le reporting avec l'analyse et la diffusion en mode zéro-client, à partir de n'importe quelle source de données multidimensionnelle ou relationnelle.

Hummingbird[5]

Hummingbird BI est une puissante solution décisionnelle permettant de déployer des fonctionnalités d'interrogation, de reporting et de traitement OLAP à l'échelle de toute l'entreprise pour un coût de possession minimal. En effet, Hummingbird BI fournit l'ensemble des outils d'interrogation et de reporting nécessaires pour localiser, partager, gérer, publier et analyser l'information, ce qui permet aux utilisateurs de prendre rapidement des décisions parfaitement fondées. Hummingbird BI couvre l'intégralité des besoins utilisateur à travers une série de quatre produits:

[3]http://www.cognos.com/
[4]http://www.crsystaldecision.com/fr
[5]http://www.hummingbird.com/fr

– BI Query : application d'interrogation/reporting ad hoc pour l'entreprise qui permet à l'utilisateur d'interroger les données et d'obtenir des résultats sous la forme de rapports à haute valeur informative.
– BI Web : solution web délivrant des capacités complètes d'interrogation, de reporting et d'analyse OLAP.
– BI Analyze : application OLAP (On-Line Analytical Processing) sur plateforme PC permettant aux utilisateurs de résoudre les questions métier les plus complexes à travers l'analyse multidimensionnelle des données.
– BI Server : serveur d'applications d'entreprise conçu pour délivrer des services de sécurité, planification, distribution, notification et administration centralisée, facilitant ainsi le déploiement des fonctionnalités décisionnelles à l'échelle de l'entreprise tout entière.

IBM[6]

L'offre décisionnelle de IBM est appelée *IBM DB2*. Elle permet de répondre de façon optimale aux problématiques actuelles des entreprises en Business Intelligence et aussi dans les domaines de la Gestion de Contenu, les ERP[7] et la Gestion de la Relation Client (CRM).

L'offre DB2, complétée par celle d'Informix, propose des solutions de gestion de bases de données optimisées pour supporter les applications décisionnelles ou de business intelligence. Afin de permettre à l'utilisateur d'analyser les données métier et les mettre en forme (reporting, data mining) pour qu'il ait une meilleure maîtrise des données et des informations. Cette offre est composée des modules suivants :

– DB2 Warehouse Manager : permet l'extraction, la transformation, le mouvement et le chargement des données ainsi que gestion des méta-données. *Ascential Software* avec son offre d'intégration de données (ETL et Qualité de Données) vient enrichir cette offre.
– Red Brick Warehouse : permet les fonctions de design, conception et administration des Data Warehouses et des Datamarts.
– DB2 Olap Server : permet le développement d'applications analytiques destinées à des analyses multi-dimentionelles d'entreprise grâce à l'utilisation de fonctions financières, statistiques et mathématiques. DB2 Olap Server repose sur la technologie de Hyperion Essbase.
– QMF : répond aux besoins de requêtes et de reporting. A cela, s'ajoutent les offres de Business Objects et de Brio.

A ces modules s'ajoutent un ensemble d'outils qui permettent de réaliser des algorithmes de fouille de données comme DB2 Intelligent Miner.

[6]http://www-306.ibm.com/software/data/db2/db2olap/
[7]Enterprise Resource Planning.

Isoft

Isoft est une suite d'outils dédiés à l'analyse des données, nous citons ici les deux principaux outils de Isoft : AMADEA et ALICE:

- AMADEA. Amadéa est un outil d'ETL et de data morphing. Cet outil permet l'alimentation, la transformation (morphing) et le reporting en temps réel par la création et l'exécution de scripts. L'outil délivre des indicateurs métiers paramétrables. Amadéa construit également le système d'information support, et les data marts. Ses principales fonctionnalités se résument en : extraction, nettoyage, transformation, chargement et reporting. Ses différents modules sont au nombre de quatre : Amadéa Studio: data morphing, Amadéa Batch processing: automatisation des traitements, Amadéa Web: génération des rapports dans un navigateur et Amadéa Web mining: indicateur de CRM[8] pré-établis.
- ALICE. Alice est l'outil de data mining d'Isoft, il est complet et interactif. Il permet les différentes techniques de fouille de données suivantes : arbres de décision, analyse interactive, profiling, corrélation, clustering, segmentation.

Microsoft[9]

Microsoft propose sa solution décisionnelle dans l'outil *SQL Server OLAP Services*. Ce composant répond aux défis d'une mise en œuvre OLAP :

- La construction du modèle de données OLAP, une interface utilisateur intuitive améliore l'accessibilité aux données,
- l'explosion des données avec l'agrégation, des choix de stockage souples pour une meilleure utilisation des ressources, préagrégation intelligente, performances et évolutivité,
- la présentation des informations OLAP à l'utilisateur, mode déconnecté et transmission par le Web avec l'outil *Microsoft PivotTable Service*
- la présence des outils OLAP, l'outil *MicroStrategy 7i* qui contient un moteur ROLAP.

Oracle[10]

Après le lancement d'*Oracle Database 10g OLAP* (anciennement ORACLE EXPRESS), Oracle s'est positionné comme un des acteurs majeurs de la BI, offrant un outil tout en un. En effet cette solution a pour vocation d'éviter aux clients l'achat de plusieurs outils serveurs spécialisés entre autres dans l'extraction, le datamining, les bases de données et les bases de données OLAP. La suite logicielle de Business Intelligence "Oracle Business Intelligence 10g" a permis à Oracle d'étendre sa philosophie de tout en un pour la partie BI client, et est en mesure d'assurer l'automatisation du processus en supportant toutes les étapes

[8] Gestion des relations clients.
[9] http://www.microsoft.com/france/technet/produits/sql/7.0/a-olap-1005.mspx
[10] www.oracle.com/olap

de la prise de décision. Parmi les composants de cette solution, un moteur multidimensionnel très puissant et un outil de construction d'entrepôt de données appelé *warehouse builder*.

SAS[11]

Avec ses deux outils SAS9 et SAS Miner la plateforme décisionnelle SAS est une solution de BI complète elle contient différentes fonctionnalités les principales sont : intégration de données (ETL, qualité de données...), stockage, métadonnées uniques, portail web, reporting de masse, interactif ou non, analyse de type OLAP, analyse prédictive, datamining, textmining, applications métiers (marketing, ressources humaines, achats, grande distribution, finance, risque...) et pilotage stratégique.

[11] http://www.sas.fr/

ANNEXE C

Rappels sur la théorie des sous-ensembles flous.

Dans cette annexe nous allons présenter quelques éléments de base de la théorie des sous-ensembles flous qui faciliteront la compréhension du principe du système SAINTETIQ vu qu'il se base sur la théorie des sous-ensembles flous.

Sous-ensembles flous

La théorie des sous-ensembles flous proposée par Zadeh en 1965 [105] comme une extension de la théorie des sous-ensembles permet d'exprimer une appartenance graduelle reflétant un degré de satisfaction de cette appartenance. Ce qui permet d'exprimer plus facilement des situations plus proches du langage humain qui peuvent être aussi des informations imprécises ou des classes aux limites mal définies. Si par exemple on veut qualifier une personne de *jeune*, ça peut être exprimé sous la forme $Jeune = \{x \in P, 18 \leq x \leq 25\}$. Ici P est l'ensemble des personnes, cet exemple qualifie une personne de 17 ans comme une personne *non jeune*, alors que cette personne peut être qualifiée de *jeune*. A l'aide des sous-ensemble flou on pourrait présenter ce genre d'information.

Definition C.1 (Sous-ensemble flou)**.** *Un sous-ensemble flou A est défini sur un ensemble de référence ou référentiel Ω par une fonction d'appartenance μ_A de Ω vers l'intervalle réel $[0, 1]$:*

$$\begin{array}{rccc} \mu_A : & \Omega & \rightarrow & [0,1] \\ & x & \rightarrow & \mu_A(x) \end{array}$$

La figure C.1 montre une représentation d'un sous-ensemble flou où la fonction d'appartenance μ_A qui nous permet de connaître l'appartenance d'un individu qui a un âge donné à cette classe (ensemble flou) *jeune*. Pour un ensemble classique, la fonction d'appartenance ne prend plus ses valeurs dans l'intervalle $[0, 1]$ mais dans l'ensemble $\{0, 1\}$. Un ensemble classique est donc un cas particulier de sous-ensemble flou. Mais il reste possible de dériver d'un sous-ensemble flou un ensemble classique, dénommé α-coupe, répondant alors à l'une des définitions suivantes :

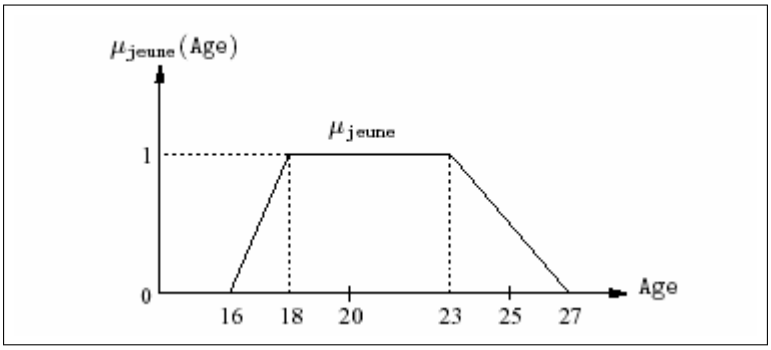


Figure C.1 – Représentation du sous-ensemble flou *Jeune*

Definition C.2 ($\alpha-coupe$)**.** *On appelle $\alpha-coupe$ (ou coupe de niveau α ou encore sous-ensemble de niveau α) d'un sous-ensemble flou A du référentiel Ω, et pour toute valeur $\alpha \in [0,1]$[1], l'ensemble classique A_α défini par :*

$$A_\alpha = \{\mu \in \Omega, \mu_A(\mu) \geq \alpha\}$$

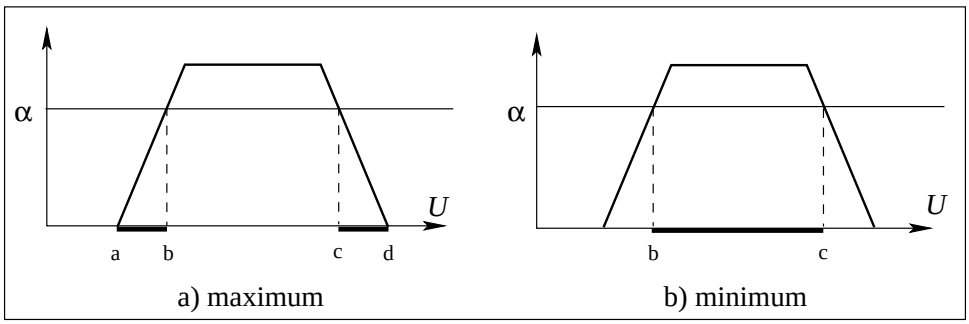


Figure C.2 – $\alpha-coupe$

Definition C.3 (α-coupe stricte)**.** *On appelle $\alpha-$ coupe stricte pour une valeur $\alpha \in [0,1]$, d'un sous-ensemble flou A du référentiel Ω ,l'ensemble classique $A_{\bar{\alpha}}$ défini par :*

$$A_{\bar{\alpha}} = \{\mu \in \Omega, \mu_A(\mu) \geq \alpha\}$$

Caractéristiques des sous-ensembles flous.

Soit A un sous-ensemble flou sur un ensemble Ω.

– Support : le support de A, noté *supp*, est l'ensemble (*classique=)* des éléments de Ω qui satisfont le concept représenté par A.

$$supp(A) = \{x \in \Omega / \mu_A(x) > 0\}$$

– Noyau : le support de A, noté *noy*, est l'ensemble (*classique=)* des éléments de Ω qui satisfont pleinement le concept représenté par A.

$$noy(A) = \{x \in \Omega / \mu_A(x) = 1\}$$

– Hauteur : la hauteur d'un sous-ensemble flou A notée $H(A)$, la plus grande des valeurs que prend la fonction d'appartenance dans l'intervalle $[0, 1]$:

$$H(A) = sup_{u \in U} \mu_A(u)$$

On notera qu'un sous-ensemble flou est habituellement dit normalisé en rapport avec sa hauteur, lorsqu'elle vaut 1, ce qui équivaut également à un noyau non vide.

– Cardinalité : on appelle cardinalité d'un sous-ensemble flou A la valeur notée $|A|$ servant à évaluer le degré global auquel les élements de l'ensemble de référence appartiennent à A :

$$|A| = \sum_{u \in U} \mu_A(u), \ \textit{sur un référentiel discret.}$$

$$|A| = \int_{u \in U} \mu_A(u), \ \textit{sur un référentiel continu.}$$

– Spécificité : la spécificité d'un sous-ensemble flou A se détermine par rapport à un autre sous-ensemble flou B défini sur le même ensemble de référence. On dit que A est plus spécifique que B si :
  – $noy(A) \subset noy(B)$ et,

  – $noy(A) \subseteq noy(B)$

  Les sous-ensembles les plus spécifiques alors sont les singletons $\{u\}$ de U tels que :
  – $\mu_u(u) = 1$, et

  – $\forall v \in U / v \neq u, \mu_u(v) = 0$

Un modèle multidimensionnel pour un processus d'analyse en ligne de résumés flous

Lamiaa NAOUM

Résumé

Le travail présenté dans cette thèse traite de l'exploration et de la manipulation des résumés de bases de données de taille significative. Les résumés produits par le système SAINTETIQ sont des vues matérialisées multi-niveaux de classes homogènes de données, présentées sous forme de collections d'étiquettes floues disponibles sur chaque attribut. La contribution de cette thèse repose sur trois points. En premier lieu nous avons défini un modèle de données logique appelé *partition de résumés*, par analogie avec les cubes de données OLAP, dans le but d'offrir à l'utilisateur final un outil de présentation des données sous forme condensée et adaptée à l'analyse. En second lieu, nous avons défini une collection d'opérateurs algébriques sur l'espace multidimensionnel des partitions de résumés. Ces opérateurs sont à la base d'une algèbre de manipulation des résumés. Cette algèbre prend en compte les spécificités du modèle de résumé que nous traitons. Nous avons adapté la majorité des opérateurs d'analyse proposés dans les systèmes OLAP. Ainsi, nous avons identifié : les opérateurs de base issus de l'algèbre relationnelle, les opérateurs de changement de granularité et les opérateurs de restructuration. Ces résultats offrent de nouvelles perspectives pour l'exploitation effective des résumés dans un système décisionnel. Finalement, pour compléter ce travail, nous nous sommes intéressés à la représentation des résumés et des partitions de résumés linguistiques, notamment pour en fournir une présentation claire et concise à l'utilisateur final. Appliquée à une hiérarchie de résumés produite par le système SAINTETIQ, l'approche tente de construire des prototypes flous représentant les résumés.

Mots-clés : Résumés de bases de données, cubes OLAP, concepts multidimensionnels flous, aide à la décision, prototypes flous.

Abstract

The work presented in this thesis deals with the subject of exploration and manipulation of database summaries with significant size. The summaries produced by SAINTETIQ system are multilevel materialized views of homogeneous data clusters, presented with a collections of fuzzy labels available on each attribute. Our thesis contribution is based on three points. Initially we defined a logical data model called *summaries partition*, by analogy with OLAP datacubes, with the aim of offering to the end-user a tool for data presentation in condensed form and adapted to the analysis. Secondly, we defined a collection of algebraic operators on the multidimensional space of summaries partitions. These operators are the base for an algebra for handling summaries. This algebra takes into account specificities of the summary model we deal with. We adapted the majority of the operators of analysis proposed in OLAP systems. Thus, we identified: core operators resulting from the relational algebra, operators of changing granularity and operators of reorganization. These results offer new prospects for the effective summaries exploitation in a decisional system. Finally, to complet this work, we were interested in the summaries and partitions representation, in particular to provide a clear and concise presentation of it to the end-user. Applied to a summaries hierarchy produced by the SAINTETIQ system, the approach tries to construct fuzzy prototypes representing the summaries.

Keywords: Databases summarization, OLAP datacubes, multidimensional vague concepts, decision support, fuzzy prototypes.